PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G11C 11/40 A1

(11) International Publication Number: WO 95/16993

(43) International Publication Date: 22 June 1995 (22.06.95)

(21) International Application Number: PCT/US94/13909

(22) International Filing Date: 13 December 1994 (13.12.94)

(30) Priority Data:
166.483 13 December 1993 (13.12.93) US

(71) Applicant: LATTICE SEMICONDUCTOR CORPORATION [US/US]; 5555 N.W. Moore Court, Hillsboro, OR 97124-6421 (US).

(72) Inventors: TSUI, Cyrus, Y.; 378 South Gordon Way, Los Altos, CA 94022 (US). SHANKAR, Kapil; 2190 San Benito Drive, Fremont, CA 94539 (US). CHAN, Albert, L.; 930 Bautista Court, Palo Alto, CA 94303 (US).

(74) Agents: KWOK, Edward, C. et al.; Skjerven, Morrill, MacPherson, Franklin & Friel, Suite 700, 25 Metro Drive, San Jose, CA 95110 (US).

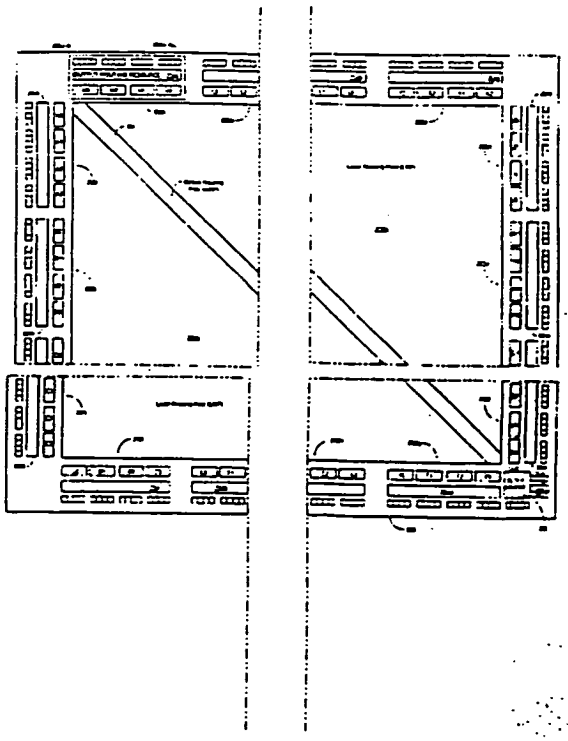
(81) Designated States: DE, GB, JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: APPLICATION SPECIFIC MODULES IN A PROGRAMMABLE LOGIC DEVICE

(57) Abstract

A structure and a method provide a programmable logic device (200) including a number of generic logic blocks (203a-203l) or more application-specific block (203b). Such application-specific block implements a specific function, such as a register file or a memory array (420). In one embodiment, the application-specific block is programmable to be either one or more single-port memory array, a first-in-first-out (FIFO) memory, or a dual port memory array. In another embodiment, the application-specific block can be configured to be a register file, a number of counters, a number of timers, or a shift register. The application-specific block can be used in conjunction with programmable logic arrays for multiplexing input and output signals into and out of the application-specific block. Interconnectivity between the generic logic blocks and the application-specific blocks, using a global routing resource, integrates into a programmable logic device, functions otherwise difficult to implement using only generic logic blocks.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo			SE	Sweden
CH	Switzerland	KR	Republic of Korea	SI	Slovenia
CI	Côte d'Ivoire	KZ	Kazakhstan	SK	Slovakia
CM	Cameroon	LI	Liechtenstein	SN	Senegal
CN	China	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
CZ	Czech Republic	LV	Latvia	TJ	Tajikistan
DE	Germany	MC	Monaco	TT	Trinidad and Tobago
DK	Denmark	MD	Republic of Moldova	UA	Ukraine
ES	Spain	MG	Madagascar	US	United States of America
FI	Finland	ML	Mali	UZ	Uzbekistan
FR	France	MN	Mongolia	VN	Viet Nam
GA	Gabon				

Application Specific Modules in a Programmable Logic Device

Background of the Invention

1. Field of the Invention

The present invention relates to the design of programmable logic devices; and, in particular, relates to the design of programmable logic devices which include application-specific modules.

2. Discussion of the Related Art

Two examples of high density programmable logic arrays are the programmable Large Scale Integration (pLSI) devices and the in-system programmable Large Scale Integration (ispLSI) devices from Lattice Semiconductor Corporation, Hillsboro, Oregon. An ispLSI device is reprogrammable in its application without being removed from the circuit board. In-system programming techniques are discussed in U.S. Patents 4,855,954 (entitled "In-system Programmable Logic Device with Four Dedicated Terminals" to Turner et al, issued August 8, 1989), 4,761,768 (entitled "Programmable Logic Device", to Turner et al, issued August 2, 1988), and 4,896,296 (entitled "Programmable Logic Device Configurable I/O Cell", to Turner et al, issued Jan 23, 1990). The in-system programming techniques discussed in these U.S. Patents are hereby incorporated by reference. Programmable logic devices can also be implemented in both volatile and non-volatile memory technologies (e.g. electrical eraseable programmable read-only memory or E^2PROM).

Figure 1 shows a block diagram of a prior art device 100, which can be implemented as either a pLSI device or an ispLSI device. As shown in Figure 1, device 100

comprises 24 generic logic blocks (GLBs) A0-A7, B0-B7, C0-C7 and D0-D7. Each GLB includes a number of input terminals, a logic array for implementing logic functions and a number of output terminals. The signals at the GLB's input terminals originate either from the routing pool 101, or directly from input/output (I/O) pins, which are shown in Figure 1 along the periphery of the device, e.g. I/O pin 102a. The signals of the output terminals of a GLB can be routed to both output routing pool 103 and routing pool 101. Output routing pool 103 routes signals between a group of GLBs and a group of I/O pins. Each I/O pin of pLSI device 100 is associated with an input/output cell ("I/O cell"), which is programmable to define whether the I/O pin is an input pin, an output pin or a bidirectional pin.

Routing pool 101 is an interconnection resource for interconnection among the GLBs. Routing pool 101 receives input signals from both the I/O pins and the output terminals of the GLBs and provides the signals received to the input terminals of the GLBs. Routing pool 101 provides connectivity between any pair of GLBs in pLSI device 100.

In Figure 1, four groups of GLBs A0-A7, B0-B7, C0-C7 and D0-D7 are shown. Each group of GLBs, together with its output routing pool, the associated I/O cells, and the associated I/O pins, form a structure called a "macroblock". In the prior art, the signal received at each I/O cell is routed to one input terminal of routing pool 101. In addition, two additional input pins are provided per megablock to receive two additional signals into global routing pool.

In the prior art, the GLBs implement all logic functions on the programmable logic device. While the GLBs are extremely flexible, memory elements, such as registers and random access memories, cannot be readily configured in a programmable logic device. Further, many frequently encountered logic circuits, although

configurable in the GLBs, can be more efficiently provided in a custom circuit, which provides both efficiency in silicon area and speed.

Summary of the Invention

In accordance with the present invention, a programmable logic device is provided having a plurality of programmable logic blocks and one or more application-specific blocks. In such a programmable logic device, a programmable logic block typically includes a fixed number of logic gates, and a fixed number of input and output terminals coupled to input/output pins, which are configured by a configuration circuit to implement a logic function. This invention provides an application-specific circuit within the programmable logic device for performing a predetermined logic function. The application-specific circuit can have a fixed number of input and output terminals coupled to the input and output pins of the programmable logic device. In addition, the programmable logic device provides a programmable interconnection resource for programmably interconnecting the programmable logic blocks and the application specific circuit. In-system programming capability can be provided in the programmable logic array. In addition, the present invention can be implemented using either volatile or non-volatile memory technology.

In one embodiment, the programmable interconnection resource comprises a switch matrix of interconnection lines. In that embodiment, a programmable interface interconnection block is provided between the application-specific circuit and the programmable interconnection resource. The programmable interface interconnection block multiplexes each of the input and output terminals of the application-specific circuit onto one or more interconnection lines of the switch matrix. The application-specific circuit further includes a gate array

to programmably multiplex input signals received from the switch matrix.

In accordance with one aspect of the present invention, the application-specific circuit includes a configuration circuit to allow the application-specific circuit to be configured into one of numerous configurations. In one embodiment, the application-specific circuit includes a dual-port memory array, which is configurable as to both word size and total array size. In that embodiment, the memory array can be programmably subdivided into one or more memory arrays. A control circuit in that embodiment provides for accessing the memory words of the dual-port memory array in a first-in-first-out manner. The control circuit includes read and write counter for generating addresses for accessing the memory array. A status signal logic circuit indicates qualitatively the amount of valid data in said memory array.

In another embodiment, the application-specific circuit comprises a plurality of registers of programmable sizes. These registers can be independently configured to have parallel or serial input data signals, parallel or serial input data signals, or other variations of these input and output data signals. Alternatively, the application-specific circuit in that embodiment can be configured to implement (i) a shift register of programmable sizes, one or more counters, or one or more timers.

The present invention is better understood upon consideration of the detailed description and the accompanying drawings.

Brief Description of the Drawings

Figure 1 shows a prior art programmable logic device having a plurality of generic logic blocks (GLBs) for configurating logic circuits.

Figure 2a shows an embodiment of the present invention in which a programmable logic device 200 is provided with application-specific modules 209a and 209b, in accordance with the present invention.

Figure 3 shows global routing pool (GRP) 201 interconnecting local routing pools (LRPs) 202a and 202b.

Figure 4a is a block diagram of an ASB 400 suitable for implementing ASB 209a of Figure 2a.

Figure 4b is a block diagram of memory circuit 420 of Figure 4a.

Figure 4c shows AND arrays 470 and 471 providing the two groups A0 and A1 of interconnect lines routing signals from an LRP to ASB 400 under either an SRAM configuration or a FIFO memory configuration of ASB 400.

Figure 4d shows an interface interconnection block (IIB) 490 routing the output signals of ASB 400 into an LRP.

Figure 5a is a block diagram of an ASB 500 suitable for implementing ASB 209b of Figure 2a.

Figures 5b(i)-5b(vi) show the various possible configurations of ASB 500.

Figure 5c shows AND arrays 570 and 571 providing the two groups A0 and A1 of interconnect lines routing signals from an LRP to ASB 500 under.

Figure 5d shows an interface interconnect block (IIB) 590 routing the output signals of ASB 500 into an LRP.

Figure 6a shows a circuit 600 suitable for implementing one of ASB 500's register banks 0, 2, 4, and 6.

Figure 6b shows a circuit 650 suitable for implementing one of ASB 500's register banks 1, 3, 5, and 7.

Detailed Description of the Preferred Embodiments

Figure 2a shows, in one embodiment of the present invention, a block diagram of a programmable logic device 200 including 6 "megablocks" 203a-203l and two application-specific modules ("ASBs") 203a and 203b. Each

megablock includes four generic logic blocks ("GLBs"), an output routing pool, and 16 I/O cells. In addiiton, programmable logic device 200 includes a clock distribution structure 206 (not shown), two local routing pools ("LRP") 202a and 202b, and a global routing pool ("GRP") 201. LRP 202a provides connectivity among megablock 203a-203c and the interface interconnect block ("IIB") 208, and LRP 202b provides connectivity among megablocks 203d-203f and IIB 208. GRP 201 provides connectivity between LRPs 202a and 202b. The hierarchical routing scheme, i.e. the use of local and global routing pools, are discussed in detail in copending U.S. Patent Application, serial no, 08/115,723, entitled "Structure and Method for Implementing Hierachical Routing Pools in a Programmable Logic Device," by C. Tsui et al, filed on September 1, 1993, Attorney Docket Number M-2491-US, assigned to Lattice Semiconductor Corporation, which is also the Assignee of the present invention. The disclosure of copending Application 08/115,723 is hereby incorporated by reference in its entirety.

Clock distribution network 206 receives five external clock signals on five dedicated clock pins. The GLBs can also be configured to generate additional signals as clock input signals to clock distribution network 206. Five clock signals (three clock signals generated in the logic blocks, e.g. GLBs or ASBs, and two clock signals received from the I/O cells) can be distributed from clock distribution network 206 to any GLB, ASB or I/O cell of programmable logic device 200.

Figure 2b shows a typical structure 250 of a GLB in programmable logic device 200. As shown in Figure 2b, each GLB has 24 input terminals (collectively referred to as input terminals 257), eight output terminals O1-O7, and an AND array 256. Input terminals 257 carry signals to AND array 256 from the LRP (say, LRP 202a) which serves GLB 250. AND array 256 receives (a) output signals of GLBs served by LRP 202a, (b) output signals from I/O cells

in the megablocks which includes GLBs served by LRP 202a, and (c) output signals received over GRP 201. The signals received over GRP 201 include output signals of GLBs served by LRP 202b, and output signals from I/O cells in the megablocks which includes the GLBs served by LRP 202b. AND array 256 also receives four internal "fast feedback" signals from the product-term sharing arrays ("PTSAs") 251a and 251b of GLB 250.

AND array 256 generates 2 groups of twenty product-term signals. Each product-term signal, which is provided both in logic true and the complement forms, is a logic product involving any number of the input signals received into AND array 256. In Figure 2b, a first group of twenty product term signals is received by PTSA 251a, and a second group of twenty product term signals is received by PTSA 251b. Product-term sharing arrays, such as product term sharing arrays 251a and 251b, are discussed in detail in U.S. Patent 5,130,574, entitled "Programmable Logic Device Providing Product Term Sharing and Steering to the Outputs of the Programmable Logic Device," to J. Shen et al, filed on May 6, 1991, issued on July 14, 1992, and assigned to Lattice Semiconductor Corporation. U.S. Patent 5,130,574 is hereby incorporated by reference in its entirety.

Output stage 252a includes output logic macrocells providing various programmable logic functional capabilities for generating either a registered or a combinatorial output signals of PTSA 251a. This type of logic macrocells used in output stage 252a are discussed in detail in U.S. Patent 5,191,243, entitled "Output Logic Macrocell with Enhanced Functional Capabilities," to J. Shen et al, filed on May 6, 1991 and issued on Mar. 2, 1993. U.S. Patent 5,191,243 is hereby incorporated by reference in its entirety.

The output signals of multiplexer 253a-253d are provided as output signals O0-O3. Output signals O0-O3 are routed back to LRP 202a or, alternatively, via the

output routing pool of GLB 250's megacell to the associated I/O cells.

Figure 2c shows output routing pool 204d, which serves megablock 203d of programmable logic device 200 (Figure 2a). As shown in Figure 2c, GLBs A0-A3 of megablock 203d are connected to a switch matrix 270. Switch matrix 270 receives from GLB A0-A3 their 32 output signals, and provides 16 output signals to I/O pins 205d-0 to 205d-15. Each of the 32 output signals of GLB A0-A3 is routed to four I/O cells. These I/O cells can each be independently configured as an input pin, an output pin, or a bidirectional pin. When configured as an input pin, the input signal received is routed via bus 271 to LRP 202b.

To provide higher speed connections, in each GLB, four output terminals are each provided a programmable bypass signal path to an I/O cell. These bypass connections are shown in Figure 2d.

LRP 202a and LRP 202b are interconnection resources serving megablocks 203a-203f and IIB 208. IIB 208 is, in turn, a routing resource for routing output signals of ASB 209a and 209b to LRP 202a, as shown in Figure 2a. Input signals to ASB 209a and 209b can be routed from LRP 202a directly into ASB 209a and 209b. Figure 3 is a block diagram of a structure 300 implementing LRP 202a, LRP 202b, and GRP 201. As shown in Figure 3, each LRP has (i) 96 input terminals, collectively indicated by reference numeral 302a or 302b, receiving input signals from the I/O cells associated with the GLBs served by the LRP, (ii) 192 input terminals, collectively indicated by reference numeral 301a or 301b, receiving input signals from the eight output signals from the ASBs and the GLBs served by the LRP, (iii) 96 input terminals, collectively indicated by reference numeral 303a or 303b, for receiving from GRP 201 signals which originate from the other LRP, and (iv) 192 output terminals, collectively indicated by reference numeral 305a or 305b, for routing each of the 288 input

signals received by the LRP to 96 of the input terminals of the ASBs and GLBs served by the LRP. LRP 202a and 202b are each implemented according to the interconnect structure discussed in U.S. Patent 5,204,556, entitled "Programmable Interconnect Structure for Logic Blocks," by Kapil Shankar, filed May 6, 1991, issued April 26, 1993 and assigned to Lattice Semiconductor Corporation, which is the Assignee of the present application. U.S. Patent 5,204,556 is hereby incorporated by reference in its entirety. LRP 202a provides 96 output terminals to route signals through GRP 201 to LRP 202b. Likewise, LRP 202b provides 96 output terminals to route signals through GRP 201 to LRP 202a.

GRP 201 interconnects LRP 202a and 202b. To improve performance, the interconnections 305a and 305b are provided buffers, which are collectively indicated by reference numerals 304a and 304b respectively. Using GRP 201 to interconnect LRP 202a and 202b maintains connectivity between the GLBs and the ASBs served by LRP 202b and the GLBs and the ASBs served by LRP 202b, while routing only a fraction of total number of signals received by LRPs 202a and 202b. Consequently, significant reduction of silicon area is achieved. Structure 300 can be optimally used by preferentially implementing interconnected logic circuits in GLBs served by the same LRP, thereby reducing the number of signals required to be transmitted between LRPs via GRP 201. Under such condition, signals between GLBs, or between a GLB and an ASB, are transmitted over an LRP, which is a smaller interconnection circuit than the alternative under a single routing pool system. Thus, despite the increase in the number of GLBs in the programmable logic device, the signal delays in transmission through the interconnection circuit are kept generally low. Because of the reduced load in the LRP, which serves a smaller number of GLBs, the AC switching currents are generally also kept low. LRP 202a and LRP 202b, and GRP 201 are each regular in

structure, so that signal delays within each LRP, or over GRP 201, remain predictable and consistent.

In the present embodiment, ASB 209a and 209b are respectively a memory module and a register module. In this embodiment, ASB 209a can be configured as one of the following: (i) a single-port static random access memory (SRAM), (ii) two single-port SRAMs, (iii) a dual-port SRAM, and (iv) a first-in-first-out (FIFO) memory. Figure 4a is a block diagram of an ASB 400 suitable for implementing memory module ASB 209a of Figure 2a. ASB 400 has a memory circuit 420 which can be accessed from two ports "A" and "B". In this embodiment, ASB 400's memory array can be organized either as a 512 X 9 or a 256 X 18 memory array.

Ports A and B of ASB 400 each comprise 18 data I/O signals, 8 address signals, a busy signal, a chip select signal, and and two read/write control signals. The address signals are primarily used when ASB 400 is configured as SRAMs. When configured as a FIFO memory, four status signals "full", "almost full", "empty", and "almost empty" are generated by ASB 400. A reset signal is also received by ASB 400 when it is configured as a FIFO memory.

Port A is accessible from the I/O pins of the programmable device. As shown in Figure 4a, I/O cells 0-17 form an 18-bit data bus accessible through 18 corresponding pins of the programmable logic device. An output enable signal, which is either generated in a GLB or received as a global output enable signal, is provided to I/O cell 0 from one of two groups A0 and A1 of interconnection lines receiving signals from an LRP (A0 and A1 are output lines of two AND arrays 470 and 471, described below, used for routing the signals lines of the LRP into ASB 400). Output enable signals for programmable logic device 200 are discussed in a copending Application, serial no. 08/115,475, entitled "Output Enable Structure and Method for a Programmable Logic Device," by Ju Shen et

al, filed on September 1, 1993, and assigned to Lattice Semiconductor Corporation. This copending Application is hereby incorporated by reference in its entirety. The output enable signal used selected for I/O cell 0 is provided by I/O cell 0 to each of the I/O cells 1-17.

In port A of ASB 400, bus 412a carries the input signals received at I/O cells 0-17 to memory circuit 420 as data input signals. Data output signals provided by memory circuit 420 are routed to I/O cells 0-17 by bus 410a. ASB 400 can also be configured to have either a 9-bit data path or a 18-bit data path. When a 9-bit data path is desired, I/O cells 9-17 can be used for other input/output purposes. A busy signal generated by ASB 400 indicates a "busy" port A, when the memory cell sought to be accessed through port A is simultaneously accessed by a pending memory access at port B. This busy signal is routed to I/O cell 18 via lead 404a. When ASB 400 is configured for use as an SRAM, I/O cells 19-26 form a 8-bit bus for receiving a memory address, which is routed to memory circuit 420 via bus 411a. When ASB 400 is configured for use as a FIFO memory, I/O cell 19 receives a reset signal which is routed to memory circuit 420 on lead 421, and I/O cells 20-23 receive the four status signals "full", "almost full", "empty" and "almost empty" from memory circuit 420 via leads 406-409 respectively. Memory circuit 420 can also receive an internally generated reset signal, which is routed from the associated LRP into ASB 400 through group A0 of interconnect lines. ASB 400 can be programmed to select either the reset signal on lead 421 or the reset signal on lead 417 routed via the LRP. When configured as an SRAM, ASB 400 receives read/write control signals and a chip select signal resectively from either (i) I/O cells 27-29, over leads 413c, 414c, and 415c, or (ii) the LRP, via leads 413a, 414a, and 415a.

The output signals of memory circuit 420, e.g. data output signals of Port B, are routed to an IIB 490, which

provides an interface into an LRP. These output signals of memory circuit 420 are the data output signals of port B (bus 410), busy signals for ports A and B (leads 404a and 404b, respectively), "full" signal (lead 406), "almost full" signal (lead 407), "almost empty" signal (lead 408), and "empty" signal (lead 409).

The I/O signals of port B of memory circuit 420 are similar to the I/O signals of port A. However, since the signals at port B are routed through the LRP, the signals at port B can be coupled to the GLBs of programmable logic device 200. Memory circuit 420 can therefore be used to support the logic functions implemented in the GLBs. Thus, memory circuit 420 provides an effective on-chip memory for many applications.

Figure 4b shows an implementation of memory circuit 420 of Figure 4a. As shown in Figure 4b, memory circuit 420 comprises a dual-port SRAM 425, which is accessed either from port A or port B. When ASB 400 is configured as an SRAM, multiplexers 440, 442, and 443 are programmed to select (i) the port A address signals of bus 452, (ii) the read/write control signal on lead 451, and a chip select signal on lead 450. The address signals on bus 452 are the signals shown in Figure 4a on bus 411a. The read/write control signal on bus 451 can be one of the read/write control signals of leads 413a, 413c, 414a or 414c of Figure 4a. The read/write control signals of Figure 4a also selects whether ASB 400 has a 9-bit data path or an 18-bit data path at the I/O pins. Selection of data path width is accomplished by multiplexer 430, which routes the data input/output signals of dual-port SRAM 425 to the 18-bit data path (busses 460 and 461), or the 9-bit data path (either bus 460 or 461, as detemined by the programming of programmable logic device 200). The chip select signal on lead 450 can be the signal on either lead 415c or 415a of Figure 4a. The port B signals under an SRAM configuration of ASB 400 are provided in a manner

similar to that described above with respect to port A signals.

When ASB 400 is configured as a dual-port SRAM, a busy signal for one port is generated whenever a later arriving memory access at that port attempts to access the same memory address as the memory address being then accessed at the other port. The busy signal is generated by arbitration logic circuit 429, which compares addresses at port A and port B whenever the chip select signals at both ports A and B are asserted. The busy signal at port A is asserted when the chip select signal at port A is later than the chip select signal at port B, and vice versa.

When ASB 400 is configured as two single-port SRAMs, each SRAM is either 128 X 18-bit or 256 X 9-bit. To divide the address space of dual-port SRAM 425 into the two single-port SRAMs each independently accessed through one of ports A and B, multiplexers 441 and 437 fixes bit 7 of the addresses presented at ports A and B to opposite values.

When ASB 400 is configured as a FIFO memory, one of ports A and B is selected to be the "read" port (i.e. output port) of the FIFO memory, and the other port is selected to be the write port (i.e. input port) of the FIFO memory. Read counter 427 generates the read address of the FIFO memory. This read address is routed by multiplexers 432 or 433 to either multiplexer 436 (port B) or multiplexer 440 (port A). Likewise, write counter 428 generates the write address of the FIFO memory. The write address of the FIFO memory is routed by multiplexers 432 or 433 to either multiplexer 436 (port B) or multiplexer 440 (port A). Under the FIFO configuration, to control read and write counters 427 and 428, the write enable signals of ports A and B are routed to flag logic circuit 426 on leads 451 and 454 respectively. Flag control logic circuit 426 detects the "full", "empty", "almost empty" and "almost full" conditions. The "almost full" and

"almost empty" conditions are each generated when the number of valid data stored in the FIFO memory reaches the programmable threshold for the respective condition.

Figure 4c shows AND arrays 470 and 471 providing the two groups A0 and A1 of interconnect lines routing signals from an LRP to ASB 400 under both an SRAM configuration and a FIFO memory configuration of ASB 400. The assignments of the interconnect lines in the groups A0 and A1 under each configuration are shown in the respective columns on the left of each AND array.

Figure 4d shows an interface interconnect block (IIB) 490 routing the output signals of ASB 400 into an LRP. As shown in Figure 4d, each input signal to IIB 490 can be routed to one of four LRP lines.

Figure 5a is a block diagram of ASB 500 which can be used to implement ASB 209b of Figure 2a. ASB 500 can be configured as (i) a register file of up to eight registers; (ii) a counter bank of up to four counters; (iii) a timer bank of up to four modulo counters ("timers"); and (iv) a shift register. ASB 500 can be configured to form a shift register of one of many lengths. When configured as a shift register, ASB 500 can be a serial input and serial output shift register, a serial input and parallel output shift register, or a parallel input and serial output shift register.

As shown in Figure 5a, ASB 500 comprises eight banks 501a-501f of registers, which can each be configured to be 4, 8, 12 or 16 bits wide. ASB 500 receives input signals on two groups A0 and A1 of interconnection lines from two AND arrays 570 and 571, and from I/O pins of programmable logic device 200. ASB 500 provides output signals to both an associated LRP, via an interface interconnection block (IIB) 590, and I/O pins of programmable logic device 200.

As shown in Figure 5a, block 502 contains a logic circuit for selecting an output enable signal to be used in the I/O cells of ASB 500. The I/O cells of ASB 500 associate selected input and output signals of ASB 500

with the I/O pins of programmable logic device 200. In this embodiment, to control the I/O cells of ASB 500 configured for output, block 502 can be configured to select an output enable signal from two global output enable signals and a product term output enable signal.

As shown in Figure 5a, I/O cell 503 receives an enable signal from an I/O pin. This enable signal, which is provided on lead 513 and routed to register banks 501a-501h by configuring multiplexer 514, parallel loads a selected register the data received on the I/O pins associated with I/O cells 505a-505p. When ASB 500 is configured for serial input under a shift register mode, I/O cell 504 receives a serial input signal at an associated I/O pin . This serial input signal, which is provided on lead 512, is routed to register banks 501a-501h by configuring multiplexer 516. I/O cells 505a-505p can be used to configured the associated I/O pins as input pins, output pins or bidirectional pins for parallel data input and/or output for ASB 500. Input signals received at the I/O pins associated with I/O cells 505a-505p are provided on internal data bus 511. Likewise, output signals provided on the I/O pins associated with I/O cells 505a-505p are received from internal data bus 509. When ASB 500 is configured to provide serial output, I/O cell 506 provides a serial output signal at an associated I/O cell. When ASB 500 is configured for as a counter or timer bank, I/O cell 507 provides up to four "count carry out" or "terminal count" signals. Under counter mode, the "count carry out" signals can be used to cascade ASBs of multiple programmable logic devices to form larger counters. Under timer mode, when a timer reaches a predetermined count value, a terminal count signal is asserted. The output signals of register banks 501a-501h (specifically, serial output signal 508, parallel data output bus 509 and count carry out signals 510) are also provided to IIB or "premux" 590 for routing to an associated LRP.

When configured for parallel input or parallel output, one of register banks 501a-501h is selected by demultiplexer 517, according to the value of a 3-bit signal received on the A0 group of interconnection lines. The A0 group of interconnection lines also provides a serial input signal and a parallel load enable signal, which can be selected, by suitably configuring multiplexers 516 and 514, in place of the serial input signal on lead 513 and the parallel load enable signal 512 received from the I/O pins. In addition, group A0 provides: (a) a "shift enable" signal on lead 518 for serial input or output, (b) under counter or timer mode, four "pre-load" signals on bus 519 for parallel loading of counters , (c) four "carry in" or "count hold" signals on bus 520 (in counter mode, "carry in" signals; in timer mode, "count hold" control signals); and (d) a product term clock signal on lead 521. Under counter mode, each asserted "carry in" signal causes an associated counter to increment or decrement in the next clock period. Under timer mode, each "count hold" signal prevents the associated timer from incrementing or decrementing the count in the timer during the next clock period.

The embodiment of Figure 5a can be configured to use one of the 5 clock signals: any one of three global clock signals, a clock signal received from an I/O cell, or a product term clock signal generated by internal logic.

Figures 5b(i)-5b(vi) show the various configurations of ASB 500. Figure 5b(i) shows ASB 500 being configured for parallel data input and parallel data output on busses 523 and 509, respectively. 3-bit select lines 524 select which of the eight 16-bit register banks is addressed. The enable signal on lead 525 enables the data on bus 523 to be written into the selected register bank. In this embodiment, the input bus 523 can be configured to provide true or complement logic format. Complement format is provided by inverted the input signals received from I/O

cells 505a-505p by multiplexer 515, or by selecting the inverted output of AND array 570.

Figure 5b(ii) shows ASB 500 being configured for parallel data input and both parallel and serial data output. Under this configuration, two or more adjacent register banks can be configured to form a shift register of any size between 20-128 bits, at 4-bit increments. Of course, if only one register bank is used, the shift register can be 4-bit, 8-bit, 12-bit or 16-bit. In this configuration, each register bank can be read or loaded in parallel in substantially the same manner as described above with respect to the configuration shown in Figure 5b(i). The register bank at the most significant end of the shift register can be configured to be a 4-bit, 8-bit, 12-bit or 16-bit shift register slice, with the shift register's remaining register banks each set to be 16-bit wide. In addition, when the shift enable signal on lead 518 is asserted, the bits in the shift register are shifted one bit per clock cycle, with the least significant bit shifted out of the shift register onto serial output lead 508. In this embodiment, priority is given to the enable signal on lead 525, i.e. if both the enable signal on lead 525 and the shift enable signal on lead 518 are asserted, parallel data load of a selected register bank results.

Figure 5b(iii) shows ASB 500 being configured for both parallel and serial data input and parallel data output. In this embodiment, two or more adjacent register banks can be configured to form a shift register of any size between 20-128 bits, at 4-bit increments. Of course, if only one register bank is used, the shift register can be 4-bit, 8-bit, 12-bit or 16-bit. The register bank at the most significant end of the shift register can be configured as a 4-bit, 8-bit, 12-bit or 16-bit shift register slice, with the shift register's remaining register banks each set to be 16-bit wide. When the shift enable signal on lead 518 is asserted, the bit on serial

data input line 523 is shifted into the shift register as the most significant bit. Parallel data read and data write are carried out in substantially the same manner as described above with respect to the configuration of Figure 5b(i). In this embodiment, as in the configuration of Figure 5b(ii), the enable signal on lead 525 has priority over the shift enable signal on lead 518.

Figure 5b(iv) shows ASB 500 configured for both parallel and serial data input and both parallel and serial data output. Under this configuration, two or more adjacent register banks can be configured to form a shift register of any size between 20-128 bits, at 4-bit increment. The register bank at the most significant end of the shift register can be configured to be a 4-bit, 8-bit, 12-bit or 16-bit shift register slice, with the shift register's remaining register banks each set to be 16-bit wide. Of course, if only one register bank is used, the shift register can be 4-bit, 8-bit, 12-bit or 16-bit. When the shift enable signal on lead 518 is asserted, the bit on lead 523 is shifted into the shift register as the most significant bit, and the least significant bit shifted out of the shift register onto lead 508 as a serial data output bit. In this embodiment, as in the configuration of Figure 5b(ii), the enable signal on lead 525 has priority over the shift enable signal on lead 518.

Figure 5b(v) shows ASB 500 being configured for counter operations. In this embodiment, counter functions are provided only in register banks 1, 3, 5 and 7. Thus, ASB 500 can be configured into four 4-, 8-, 12-, or 16-bit counters. Alternatively, two or more adjacent register banks can be configured to form a larger up/down counter of any size between 20-64 bits, at 4-bit increments. In such a larger counter, the register bank at the most significant end is configured as a 4-bit, 8-bit, 12-bit or 16-bit counter, with the larger counter's remaining register banks eacg set at 16 bits wide. In this configuration, when the enable signal on lead 525 is

asserted, the selected register bank is loaded in parallel. Alternatively, when the count preload signal on lead 519 is asserted, the selected counter loads the content of an adjacent register bank. For example, when the signal on lead 519 is asserted, the counter in register bank 1 loads in parallel the content of register bank 0. When a count "carry in" signal on bus 520 is asserted, depending on whether the counter associated with this count "carry in" signal is programmed to be an up counter or a down counter, the associated counter is incremented or decremented. Also, depending on whether the counter is programmed to be an up counter or a down counter, overflow or underflow condition in each counter is indicated in the corresponding "carry out" signal on bus 510.

Figure 5b(vi) shows ASB 500 being configured for timer operations. A timer is a counter which does not "wrap" around when the terminal count is reached. In this embodiment, only register banks 1, 3, 5 and 7 can provide with timer functions. As in the counter configuration of Figure 5b(v), two or more adjacent timers in ASB 500 can be configured to form a timer of any width between 20-64 bits, at 4-bit increments. Alternatively, each timer in ASB 500 can be individually configured as a 4, 8, 12 or 16-bit timer. In this embodiment, at programming time, a preset value can be specified for each timer. When the start signal on lead 518 is set to logic load, the preset value is loaded into the selected counter to perform the timer operations.

Figure 5c shows AND arrays 570 and 571 providing the two groups A0 and A1 of interconnect lines routing signals from an LRP to ASB 500. The assignments of the interconnect lines in the groups A0 and A1 are shown in the column on the left of each AND array.

Figure 5d shows an interface interconnect block (IIB) 590 routing the output signals of ASB 500 into an LRP. As

shown in Figure 5d, each input signal to IIB 590 can be routed to one of four LRP lines.

Figure 6a shows a circuit 600 suitable for implementing one of the register banks 0, 2, 4 and 6 in the present embodiment. As shown in Figure 6a, a 16-bit register 601 comprising sixteen D-type flip-flops provides the storage capacity in a register bank. The 16 output signals on bus 602 can be provided to both the common output leads of the register banks (i.e. bus 509 of Figure 5a) and the input leads of next higher register bank (i.e. if circuit 600 implements register bank 0, the output leads of circuit 600 provides input signals to the input leads of register bank 1). Bus 602 also provides input signals to shift logic block 610. Shift logic block 610 selects as a "serial-in" bit one of (a) the least significant bit on lead 603 from the previous register bank and (b) the signal on lead 523 (Figure 5a), which is a common serial input pin of the register banks. Under a shift operation, the serial-in bit is inserted at an appropriate 4-bit boundary to the value on output bus 602. For example, if the present register bank is configured as a 12-bit shift register, the serial bit is inserted by multiplexer 605a into bit 11 of the value on bus 602, resulting in the shifted value on bus 606. The remaining 11 bits (i.e. bits 0-11) of the value on bus 606 are obtained by shifting bits 0-11 of bus 602 using the shifters 604b, 604c, and 604d, with multiplexers 605b and 605c set to receive bits 8 and 4 of bus 602 respectively.

The input signals to register 601 are selected by multiplexer 607 from three sources: (a) bus 602, (b) bus 606 and (c) bus 522. The signals of bus 602 are selected when neither the shift enable signal on lead 518, nor the enable signal on lead 525, is asserted. The signals of bus 606 are selected when the shift enable signal on lead 518 is asserted, and the signals on bus 522 are selected when the enable signal on lead 525 is asserted. As shown in Figure 6a, gates 611 and 612 ensures that the enable

signal on lead 525 has priority over the shift enable signal on lead 518.

The clock signal for register 601 is selected from five sources by multiplexer 613. From the clock signal selected by multiplexer 613, multiplexer 614 further selects one of two phases. A global reset signal provided on lead 615 initializes register 601.

Figure 6b shows a circuit 650 suitable for implementing one of the register banks 1, 3, 5 and 7 in the present embodiment. To facilitate cross reference between circuits 600 and 650 of Figures 6a and 6b, like elements in circuits 600 and 650 are provided the same reference numerals. Thus, it will be appreciated that 16-bit register 601, shift logic 610, and multiplexers 613 and 614 for clock signal selection, are substantially identical in both circuits 600 and 650.

In circuit 650, in addition to receiving input signals from data bus 522, bus 602 and shift bus 606, register 601 can receive from two additional sources of input signals through multiplexer 651. These two additional sources of input signals are the "preload data" on bus 654, and the next count signals on bus 656. The next count signals of bus 656 are the output signals from counter logic 670. Multiplexer 662 selects for preload data bus 654 either the 16-bit data of the previous register bank's output signals received on bus 603, or a preset value configured at programming time by setting multiplexers 655. Multiplexers 655 include 16 two-to-one multiplexers, which are each independently programmed to provide a 16-bit value. The data on bus 654 is selected when the "preload" signal on lead 519a (a bit on bus 519 of Figure 5a) is asserted.

The "next count" signals on bus 656 are selected when circuit 650 is in either the counter or the timer mode. Depending upon whether multiplexer 657 selects for bus 675 the signals on bus 602, or their complement signals, count logic 670 implements either an up-counter or a down-

counter. When the count "carry in" signal on input lead 520a (a bit of bus 520, shown in Figure 5a) is asserted, incrementer 658 increments the output value from multiplexer 657, and provides on bus 656 as its output value the 16-bit next count signals. Circuit 661 detects the appropriate overflow or underflow condition (i.e. the bits on bus 675 are all "one's"), and provide a "count carry out" signal on lead 510a (a bit of bus 510 in Figure 5a). When circuit 650 is configured as a timer, the count "carry out" signal becomes the "terminal count" signal. In timer mode, when ther terminal count is reached, gate 660 selects as output signals of multiplexer 659 the signals on bus 602 as the "next count" signals.

The above detailed description is provided to illustrate the specific embodiments of the present invention and is not intended to be limiting. Numerous modifications and variations within the scope of the present invention are possible. The present invention is defined by the following claims.

Claims

1. An integrated circuit, comprising a programmable application-specific circuit.

2. An integrated circuit as in Claim 1, further comprising:

a plurality of programmable logic blocks; and

a programmable interconnection resource for interconnecting said programmable logic blocks and said application-specific circuit.

3. An integrated circuit as in Claim 2, wherein said programmable interconnection resource further comprises a switch matrix of interconnection lines and said application-specific circuit comprises a plurality of input and output terminals, said integrated circuit further comprises a programmable interface interconnection block between said application-specific circuit and said programmable interconnection resource, said programmable interconnection block multiplexes said input and output terminals of said application-specific circuit onto one or more of said interconnection lines.

4. An integrated circuit as in Claim 1, further comprises:

a plurality of input and output terminals; and

a programmable input resource for selectively coupling said input terminals of said integrated circuit to said input terminals of said application-specific circuit, so as to provide input signals to said application specific circuit.

5. An integrated circuit as in Claim 1, wherein said application-specific circuit includes a configuration circuit to allow said application-specific circuit to be configured into one of a plurality of configurations.

6. An integrated circuit as in Claim 5, wherein said integrated circuit is in-system programmable.

7. An integrated circuit as in Claim 5, wherein said integrated circuit is implemented using a non-volatile memory technology.

8. An integrated circuit as in Claim 5, wherein said integrated circuit is implemented using a volatile memory technology.

9. An integrated circuit as in Claim 5, wherein said application-specific circuit comprises a memory array.

10. An integrated circuit as in Claim 1, wherein said integrated circuit is in-system programmable.

11. An integrated circuit as in Claim 1, wherein said integrated circuit is implemented using a non-volatile memory technology.

12. An integrated circuit as in Claim 1, wherein said integrated circuit is implemented using a volatile memory technology.

13. An integrated circuit as in Claim 1, wherein said application-specific circuit comprises a programmably configurable memory array.

14. An integrated circuit as in Claim 2, wherein said application-specific circuit comprises a programmably configurable memory array.

15. An integrated circuit as in Claim 3, wherein said application-specific circuit comprises a programmably configurable memory array.

16. An integrated circuit as in Claim 13, wherein said memory array has a programmable word size selected from a plurality of word sizes.

17. An integrated circuit as in Claim 13, wherein said memory array can be programmably subdivided into one or more memory arrays.

18. An integrated circuit as in Claim 13, wherein said memory array comprises a dual-port memory array.

19. An integrated circuit as in Claim 18, further comprising a control circuit for accessing memory words in said dual-port memory array in a first-in-first-out manner.

20. An integrated circuit as in Claim 19, further comprising a status signal logic circuit for indicating qualitatively the amount of valid data in said memory array.

21. An integrated circuit as in Claim 20, wherein said control circuit comprises a read counter and a write counter for generating addresses to access said dual-port memory array.

22. An integrated circuit as in Claim 5, wherein said application-specific circuit comprises a programmably configurable memory array.

23. An integrated circuit as in Claim 22, wherein said memory array has a programmable word size selected from a plurality of word sizes.

24. An integrated circuit as in Claim 22, wherein said memory array can be programmably subdivided into one or more memory arrays.

25. An integrated circuit as in Claim 22, wherein said memory array comprises a dual-port memory array.

26. An integrated circuit as in Claim 22, further comprising a control circuit for accessing memory words in said dual-port memory array in a first-in-first-out manner.

27. An integrated circuit as in Claim 23, further comprising a status signal logic circuit for indicating qualitatively the amount of valid data in said memory array.

28. An integrated circuit as in Claim 27, wherein said control circuit comprises a read counter and a write counter for generating addresses to access said dual-port memory array.

29. An integrated circuit as in Claim 1, wherein said application-specific circuit comprises a plurality of registers.

30. An integrated circuit as in Claim 29, wherein said application-specific circuit further comprises a logic configuration circuit for configuring one of said registers into a shift register.

31. An integrated circuit as in Claim 29, wherein said application-specific circuit further comprises a logic configuration circuit for configuring one of said registers into a counter.

32. An integrated circuit as in Claim 29, wherein said application-specific circuit further comprises a logic configuration circuit for configuring one of said registers into a timer.

33. An integrated circuit as in Claim 2, wherein said application-specific circuit comprises a plurality of registers.

34. An integrated circuit as in Claim 3, wherein said application-specific circuit comprises a plurality of registers.

35. An integrated circuit as in Claim 5, wherein said application-specific circuit comprises a plurality of registers.

36. An integrated circuit as in Claim 34, wherein said application-specific circuit further comprises a logic configuration circuit for configuring one of said registers into a shift register.

37. An integrated circuit as in Claim 34, wherein said application-specific circuit further comprises a logic configuration circuit for configuring one of said registers into a counter.

38. An integrated circuit as in Claim 34, wherein said application-specific circuit further comprises a logic configuration circuit for configuring one of said registers into a timer.

39. An integrated circuit as in Claim 1, wherein said application-specific circuit comprises a plurality of counters.

40. An integrated circuit as in Claim 2, wherein said application-specific circuit comprises a plurality of counters.

41. An integrated circuit as in Claim 3, wherein said application-specific circuit comprises a plurality of counters.

42. An integrated circuit as in Claim 38, said timer coupled to receive an initial value from a source external to said integrated circuit.

43. An integrated circuit as in Claim 38, said timer coupled to receive an initial value from another of said registers.

44. An integrated circuit as in Claim 38, said timer coupled to receive a programmable load value.

45. In an integrated circuit, a method comprising the steps of:

providing a programmable application-specific circuit; and

providing a configuration circuit to allow said application-specific circuit to be configured into one of a plurality of configurations.

46. A method as in Claim 45, further comprising the steps of:

providing a plurality of programmable logic blocks; and

providing a programmable interconnection resource for interconnecting said programmable logic blocks and said application-specific circuit.

47. A method as in Claim 45, wherein said step of providing a programmable interconnection resource provides a switch matrix of interconnection lines and wherein said application-specific circuit is provided a plurality of input and output terminals, said method further comprising

the step of providing a programmable interface interconnection block between said application-specific circuit and said programmable interconnection resource, said programmable interconnection block multiplexes said input and output terminals of said application-specific circuit onto one or more of said interconnection lines.

48. A method as in Claim 47, further comprises:

providing said integrated circuit a plurality of input and output terminals; and

providing a programmable input resource for selectively coupling said input terminals of said integrated circuit to said input terminals of said application- specific circuit, so as to provide input signals to said application specific circuit.

49. A method as in Claim 45, wherein said integrated circuit is provided to be in-system programmable.

50. A method as in Claim 45, wherein said integrated circuit is implemented using a non-volatile memory technology.

51. A method as in Claim 45, wherein said integrated circuit is implemented using a volatile memory technology.

52. A method as in Claim 45, wherein said step of providing an application-specific circuit provides a programmably configurable memory array.

53. A method as in Claim 46, wherein said step of providing an application-specific circuit provides a programmably configuragle memory array.

54. A method as in Claim 52, wherein said memory array is provided a programmable word size selected from a plurality of word sizes.

55. A method in Claim 52, wherein said step of providing a programmably configurable memory array provides said memory array with the capability of being programmably subdivided into one or more memory arrays.

56. A method as in Claim 52, wherein said step of providing a programmably configurable memory array provides a dual-port memory array.

57. A method in Claim 56, further comprising the step of providing a control circuit for accessing memory words in said dual-port memory array in a first-in-first-out manner.

58. A method in Claim 56, further comprising the step of providing a status signal logic circuit for indicating qualitatively the amount of valid data in said memory array.

59. A method as in Claim 58, wherein said step of providing a control circuit provides a read counter and a write counter for generating addresses to access said dual-port memory array.

60. A method as in Claim 45, wherein said step of providing an application-specific circuit provides a plurality of registers.

61. A method as in Claim 60, wherein said step of providing an application-specific circuit further comprises the step of providing a logic configuration circuit for configuring one of said registers into a shift register.

62. A method as in Claim 60, wherein said step of providing an application-specific circuit further comprises the step of providing a logic configuration circuit for configuring one of said registers into a counter.

63. A method as in Claim 60, wherein said step of providing an application-specific circuit further comprises the step of providing a logic configuration circuit for configuring one of said registers into a timer.

64. A method as in Claim 45, wherein said step of providing an application-specific circuit provides a plurality of counters.

65. A method as in Claim 63, said timer being receiving an initial value from a source external to said integrated circuit.

66. A method as in Claim 63, said timer receiving an initial value from another of said registers.

67. A method as in Claim 63, said timer receving a programmable load value.

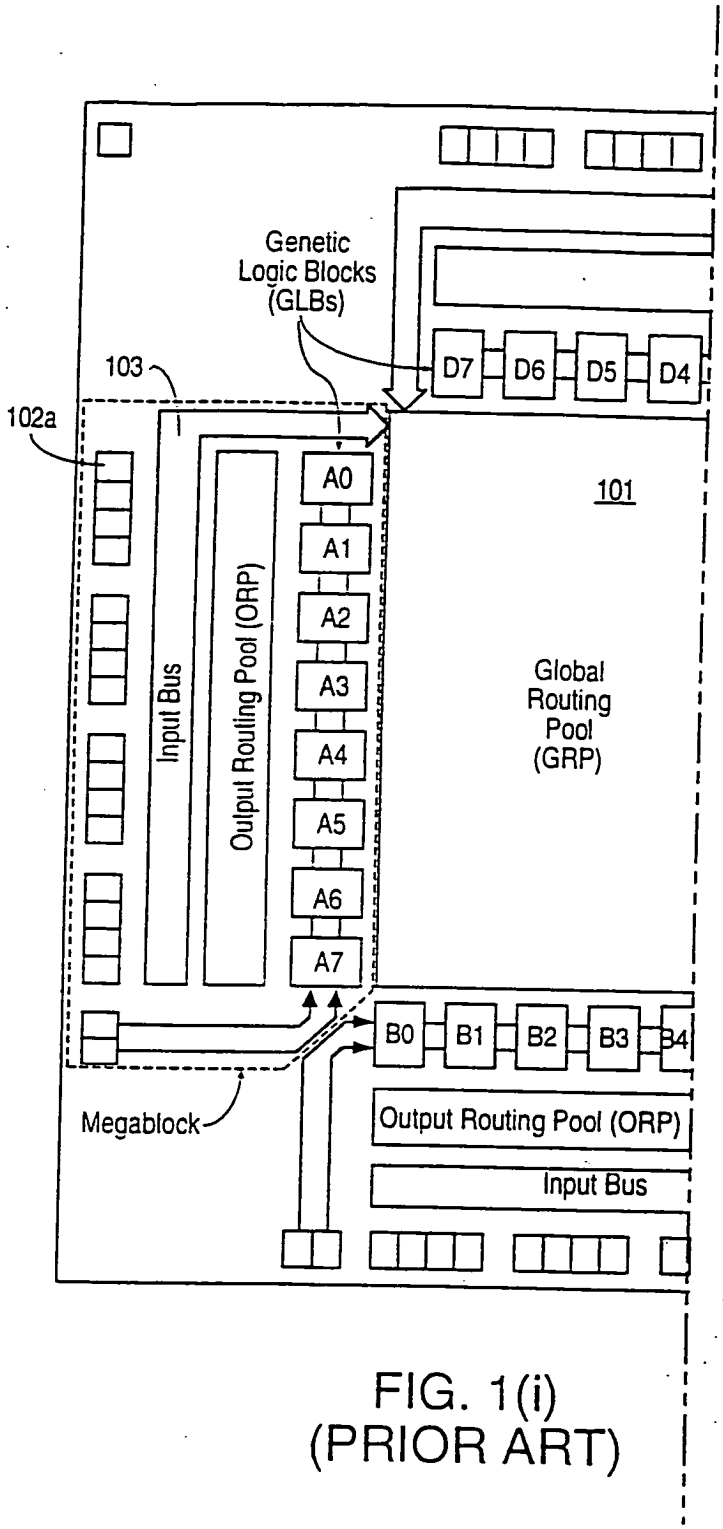


FIG. 1(i)
(PRIOR ART)

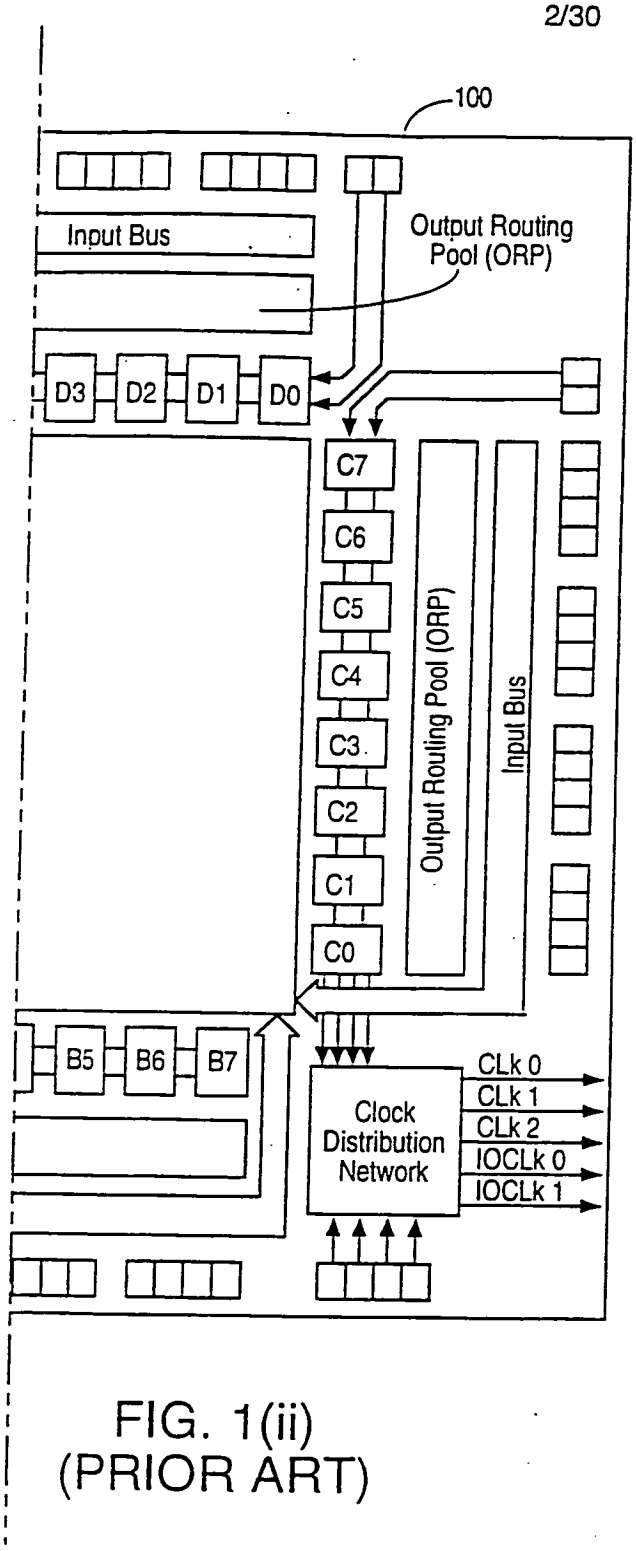


FIG. 1(ii)
(PRIOR ART)

KEY TO FIG. 1

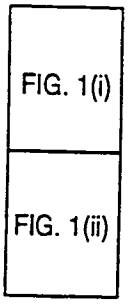


FIG. 1

FIG 2a-(i)

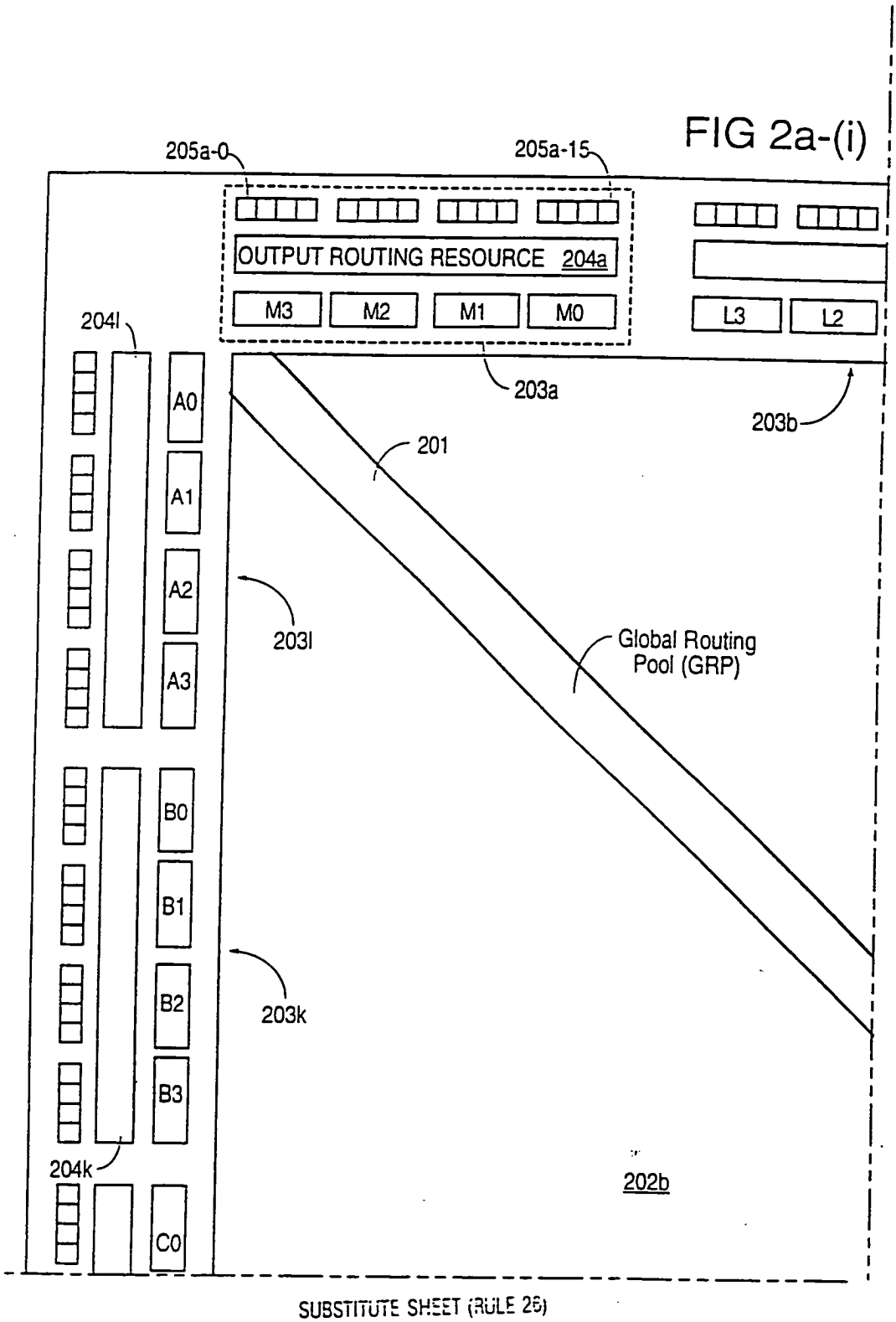
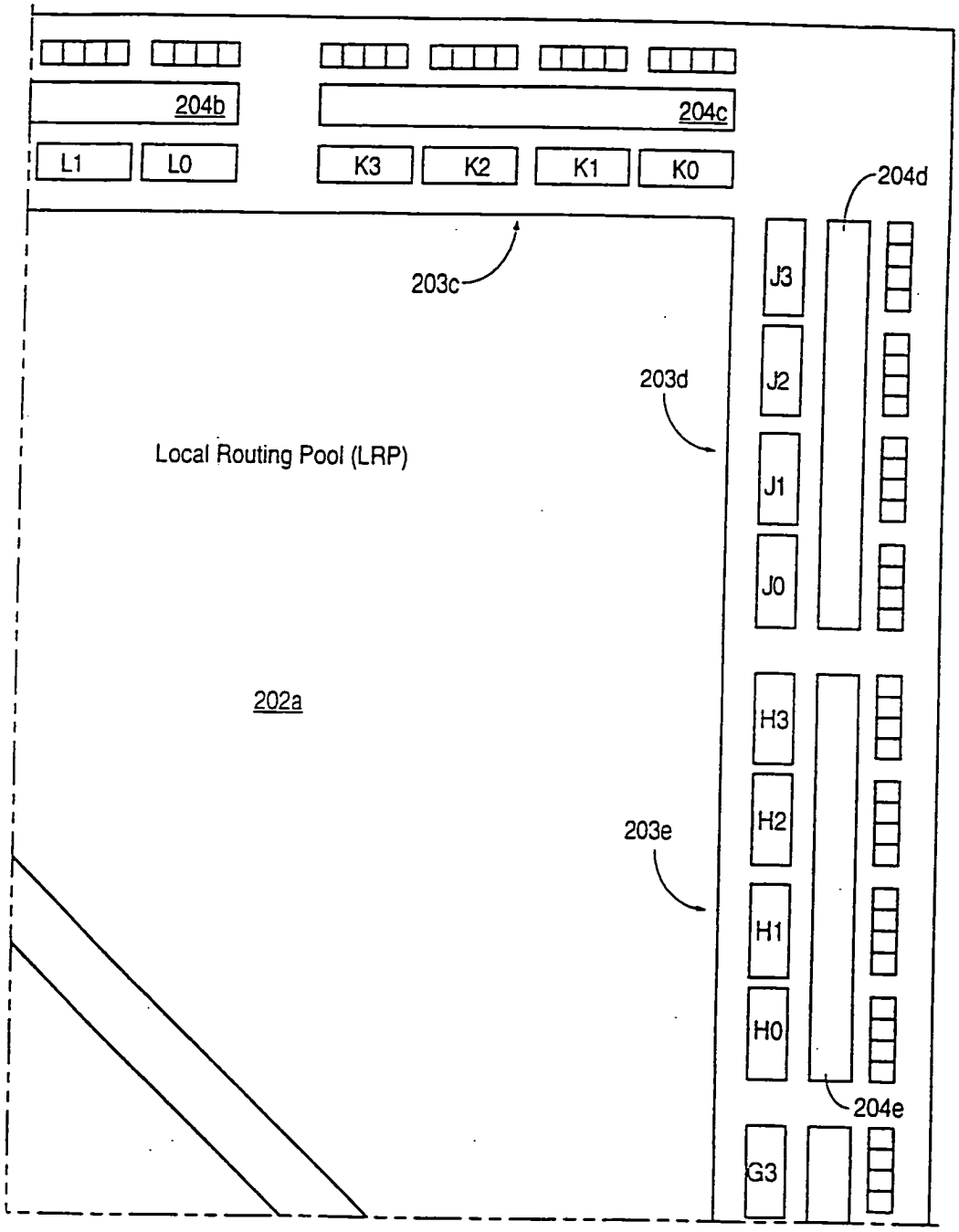

205a-0
205a-15
OUTPUT ROUTING RESOURCE 204a
M3
M2
M1
M0
L3
L2
204l
203a
203b
201
A0
A1
A2
A3
203l
Global Routing Pool (GRP)
B0
B1
B2
B3
203k
204k
C0
202b

FIG 2a-(ii)


204b
204c
L1
L0
K3
K2
K1
K0
204d
203c
J3
J2
203d
Local Routing Pool (LRP)
J1
J0
202a
H3
H2
203e
H1
H0
204e
G3

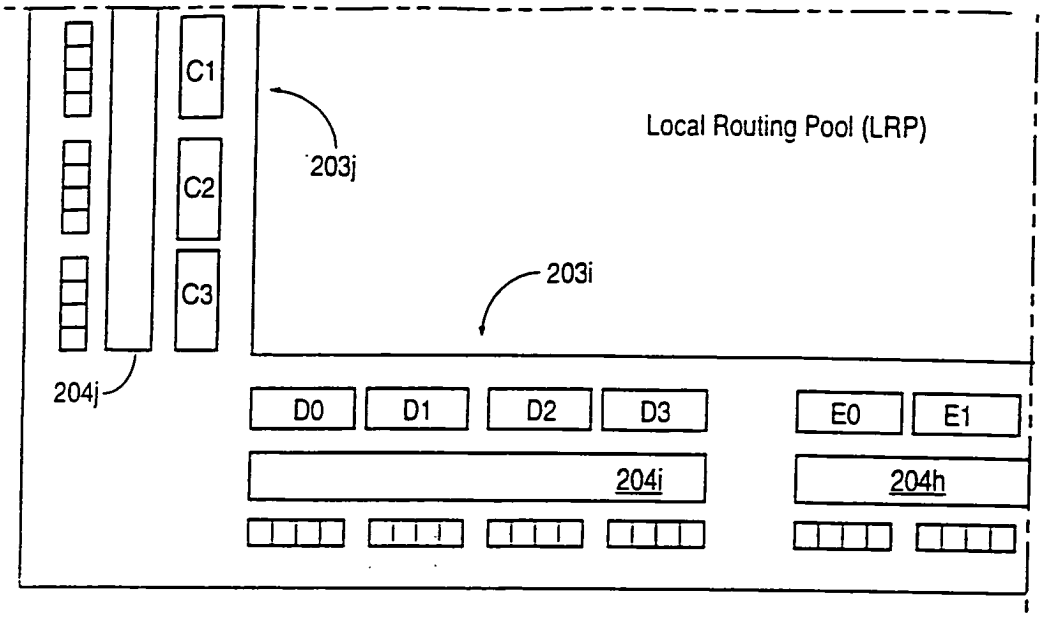


FIG 2a-(iii)

KEY TO FIG 2a

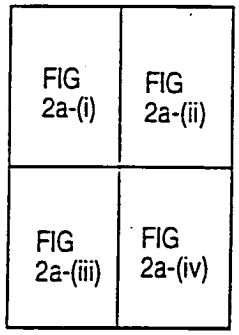


FIG 2a

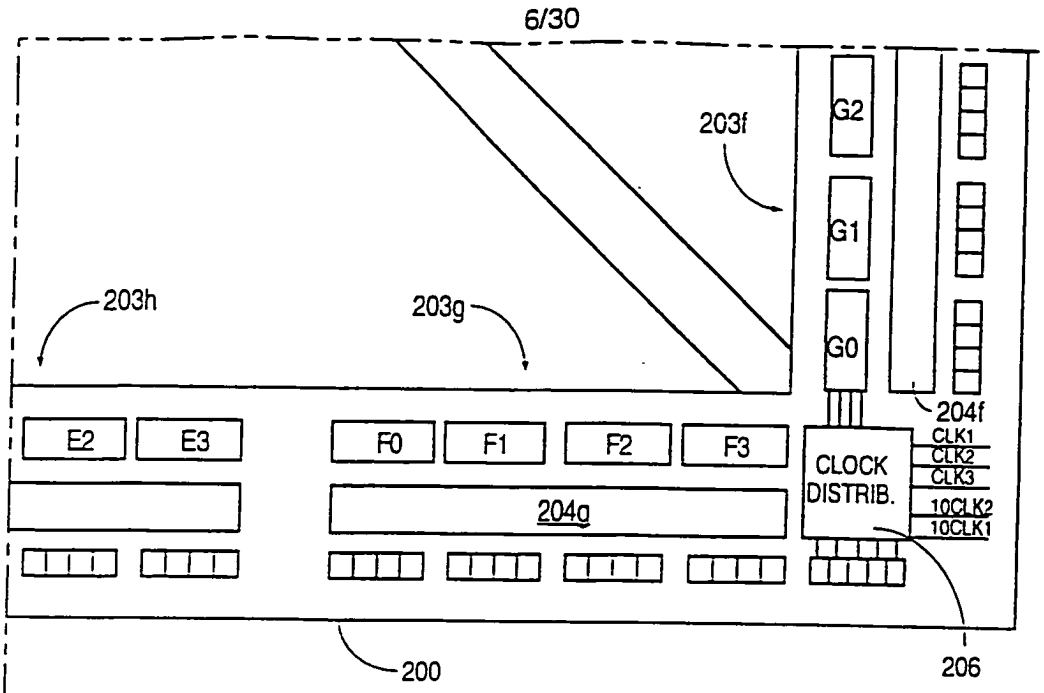

203f
G2
G1
G0
203h
203g
204f
E2
E3
F0
F1
F2
F3
CLOCK DISTRIB.
CLK1
CLK2
CLK3
10CLK2
10CLK1
204g
200
206

FIG 2a-(iv)

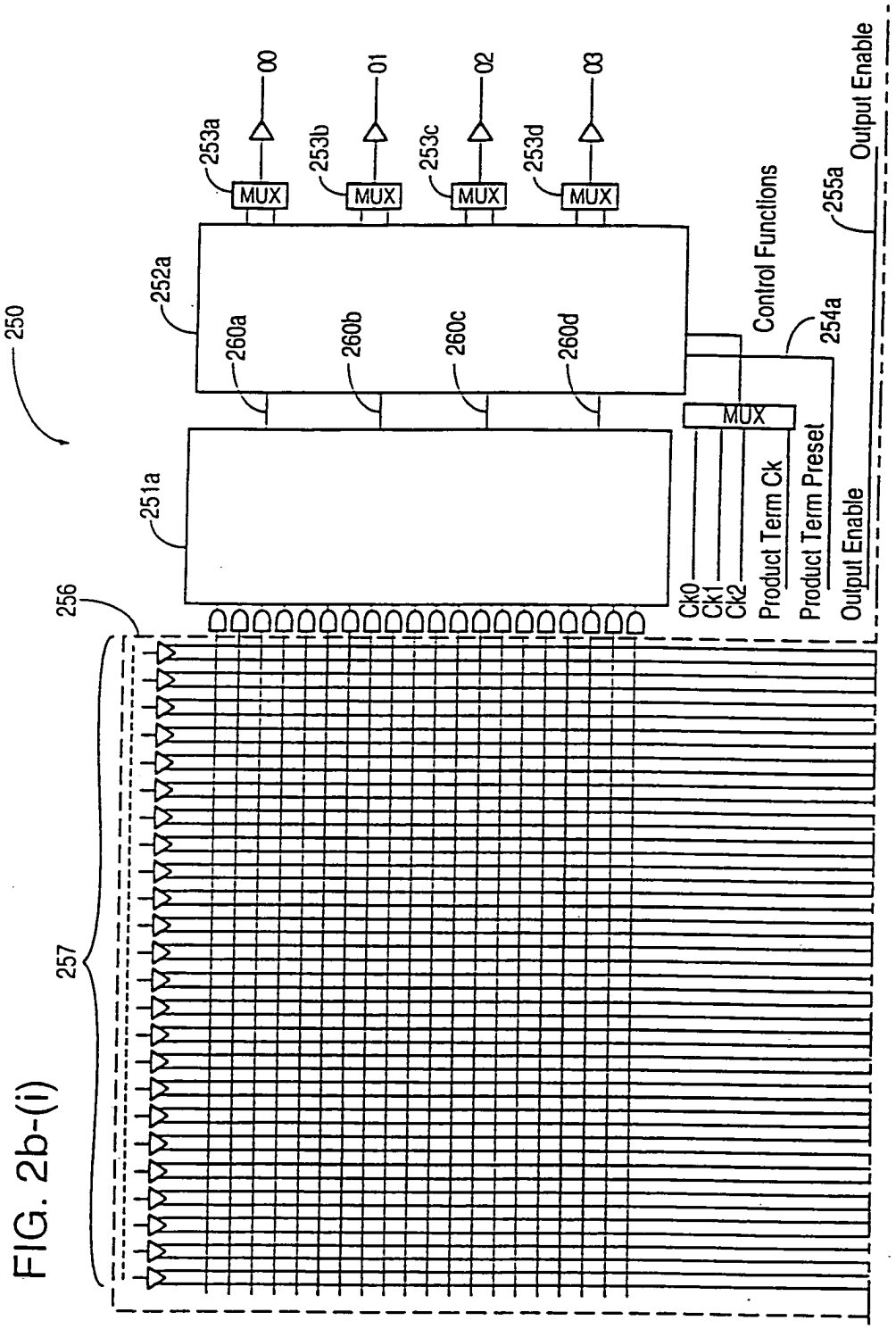

250
252a
253a
253b
253c
253d
MUX
MUX
MUX
MUX
O0
O1
O2
O3
260a
260b
260c
260d
251a
256
257
Control Functions
255a
Output Enable
254a
MUX
Ck0
Ck1
Ck2
Product Term Ck
Product Term Preset
Output Enable

FIG. 2b-(i)

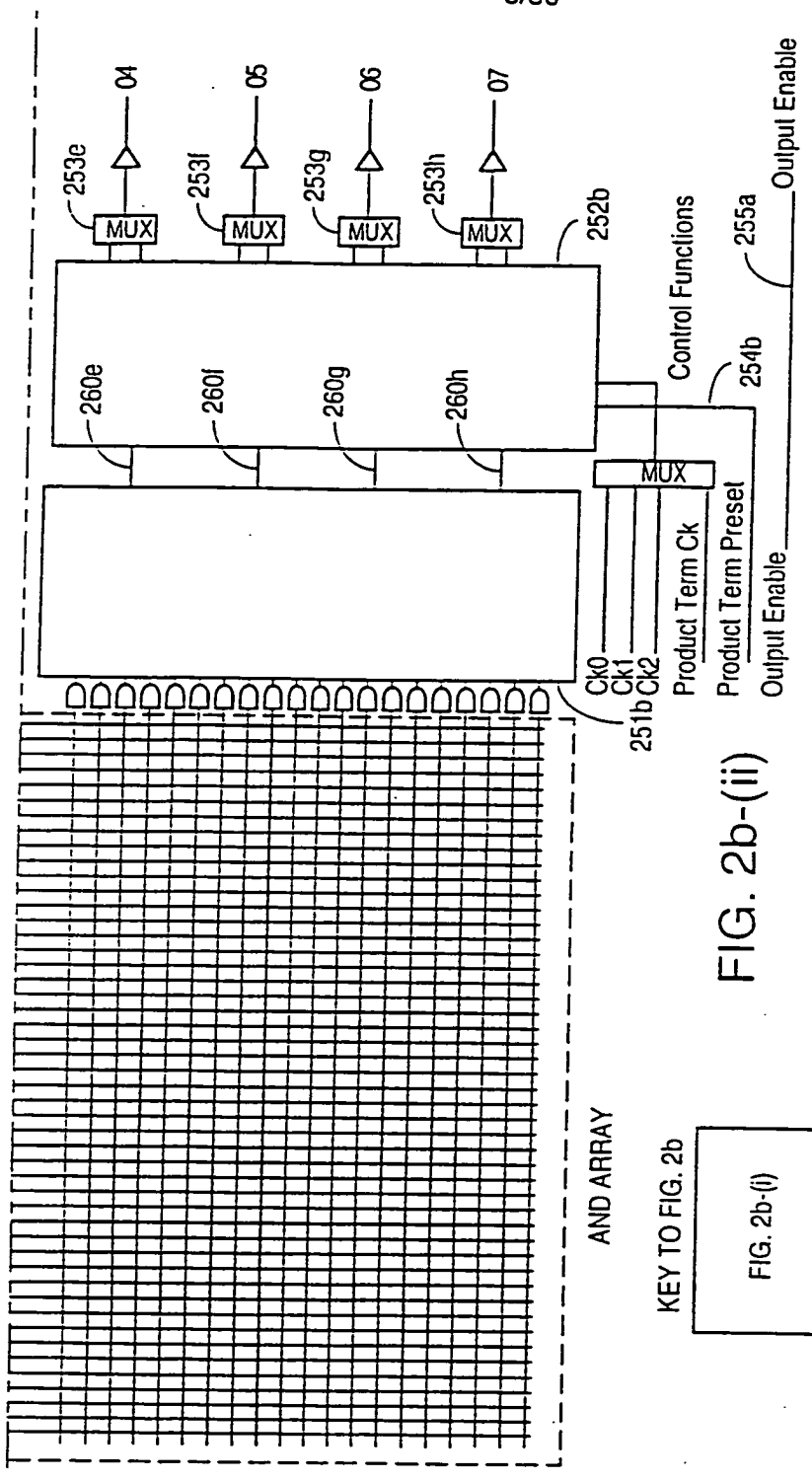


FIG. 2b-(ii)

KEY TO FIG. 2b

FIG. 2b-(i)	FIG. 2b-(ii)

FIG. 2b

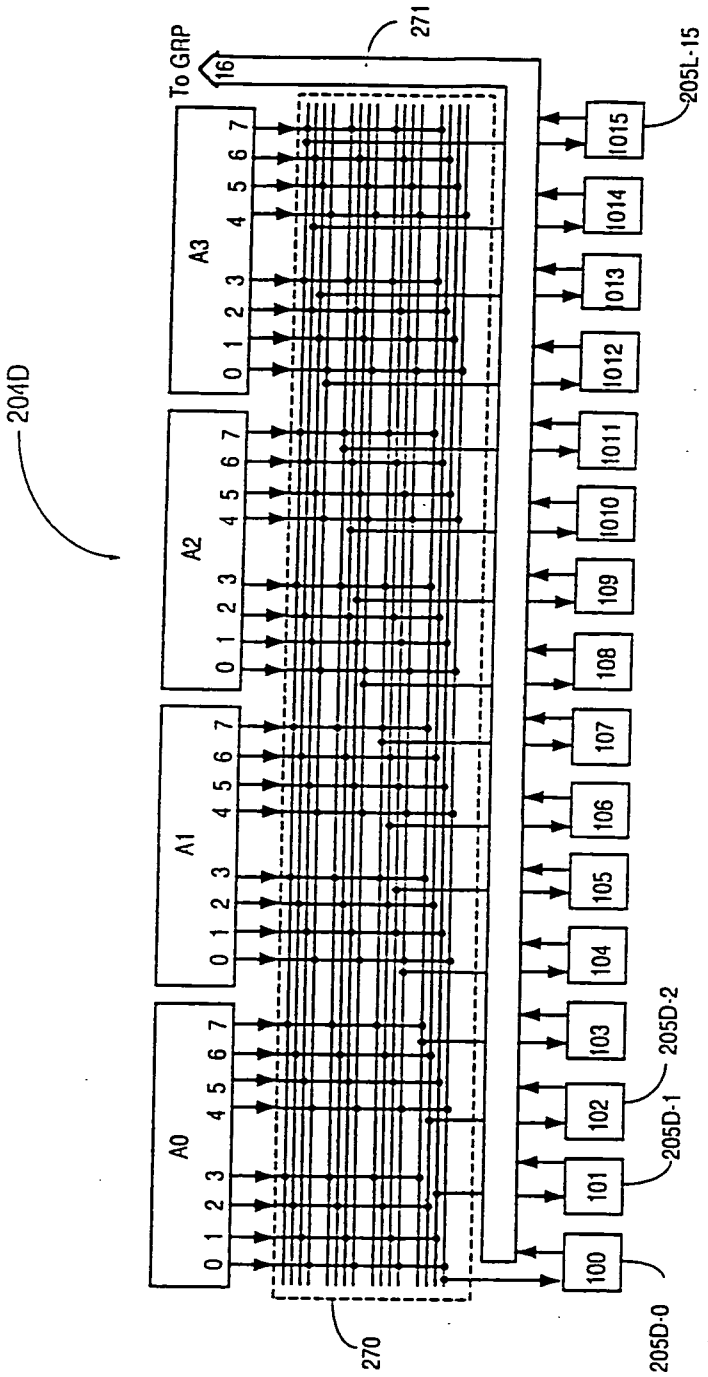

204D
271
To GRP
16
205L-15
A3
A2
A1
A0
0 1 2 3 4 5 6 7
1015
1014
1013
1012
1011
1010
109
108
107
106
105
104
103
102
101
100
205D-2
205D-1
205D-0
270

FIG. 2c

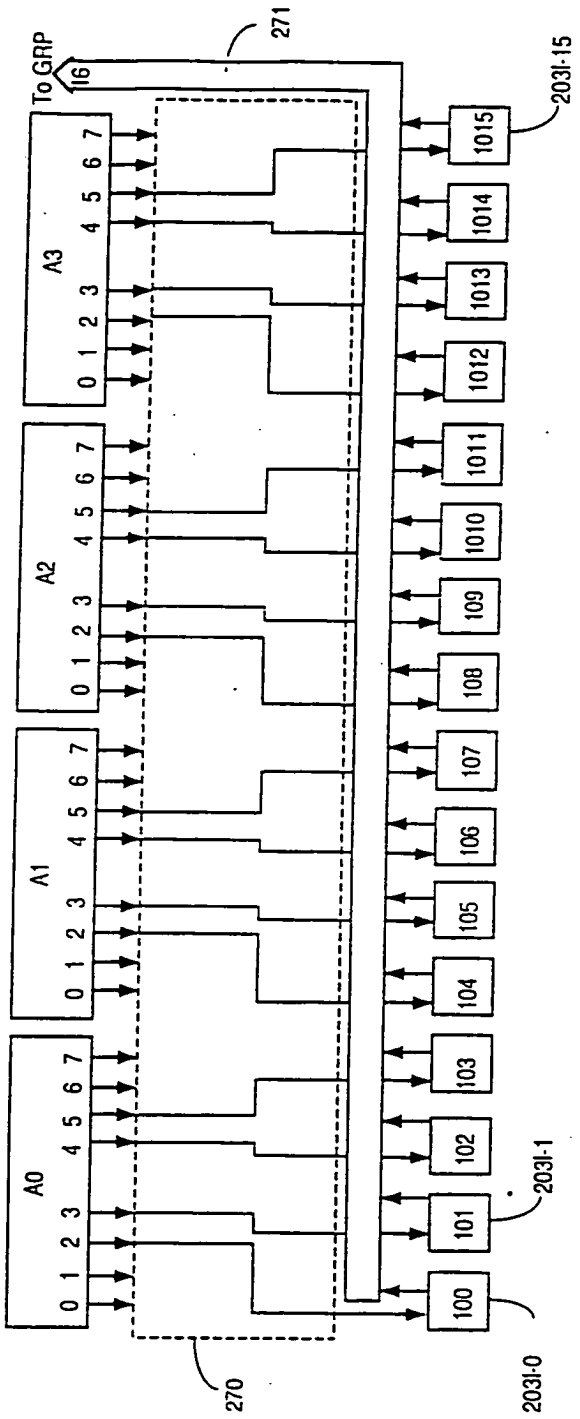

To GRP
16
271
A3
A2
A1
A0
0 1 2 3 4 5 6 7
1015
1014
1013
1012
1011
1010
109
108
107
106
105
104
103
102
101
100
203I-15
203I-1
203I-0
270

FIG. 2d

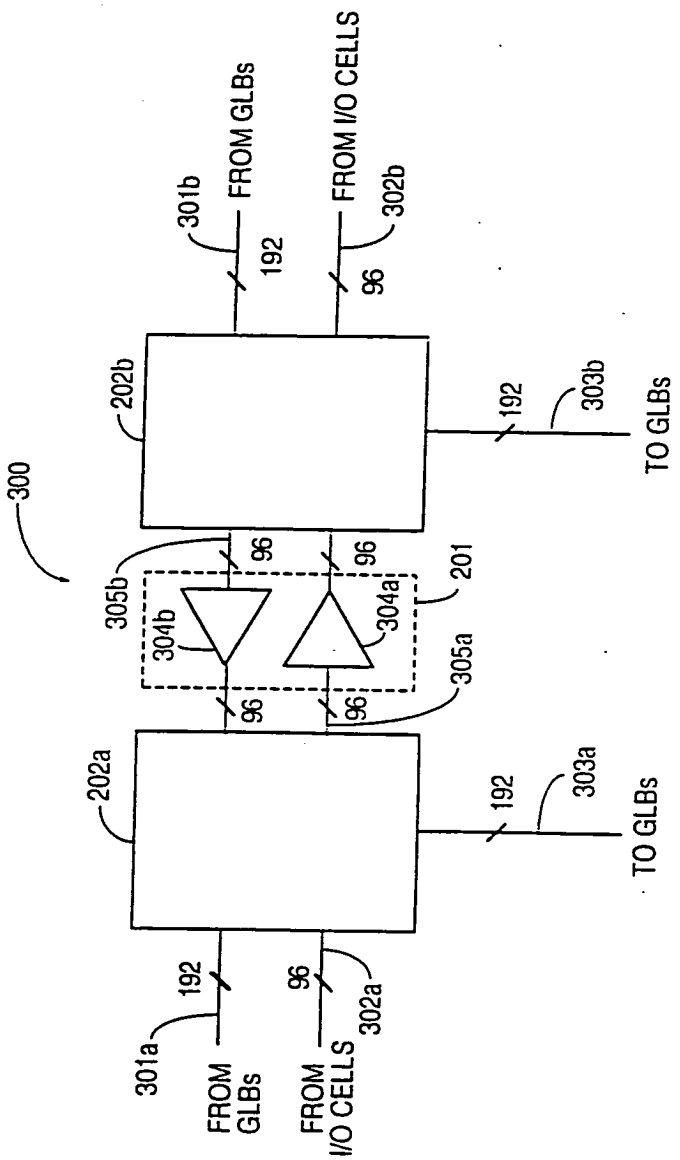

300
301b
FROM GLBs
192
FROM I/O CELLS
302b
96
202b
192
303b
TO GLBs
96
96
305b
304b
304a
201
305a
96
96
202a
192
303a
TO GLBs
192
96
301a
302a
FROM GLBs
FROM I/O CELLS

FIG. 3

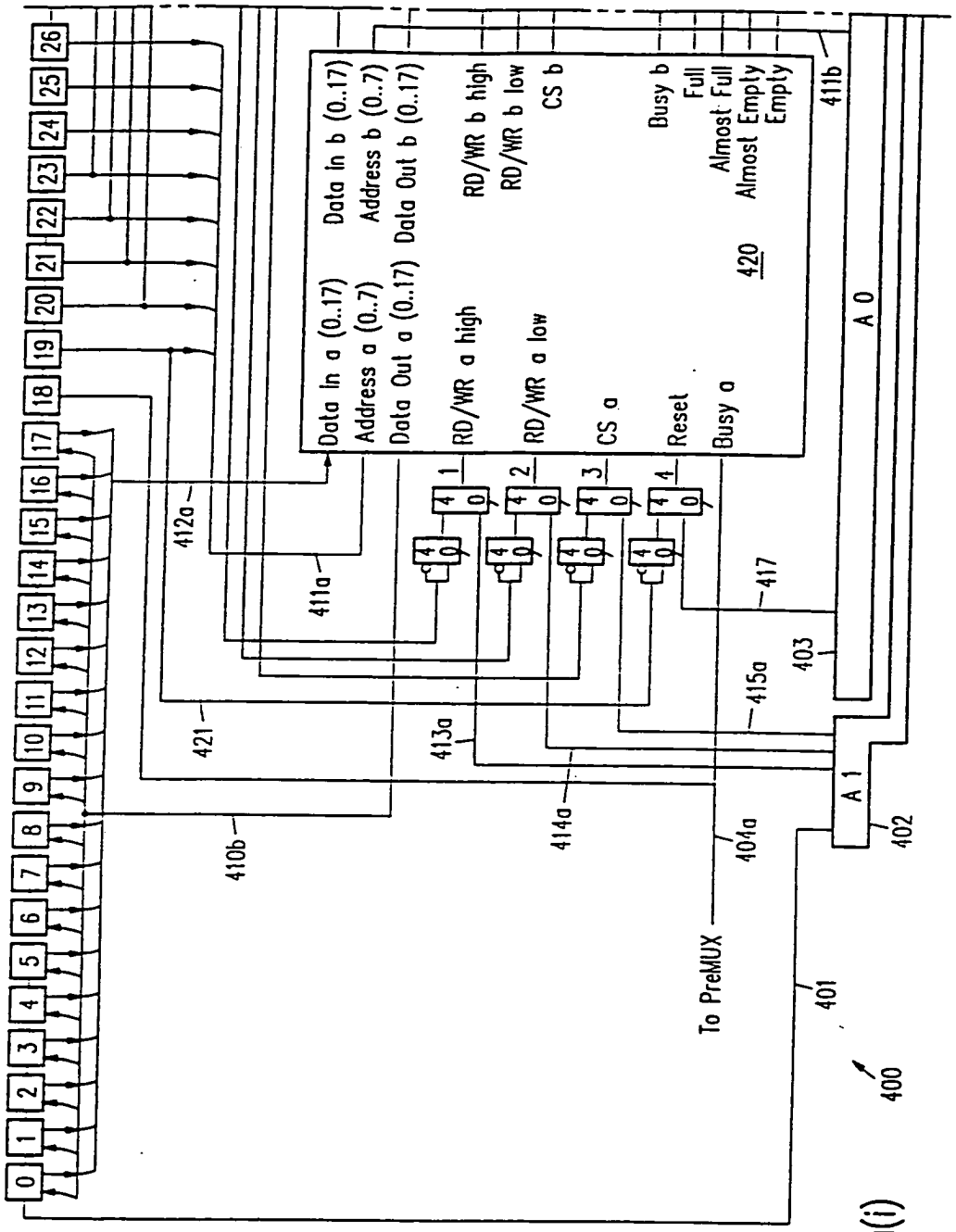

Data In a (0..17)
Address a (0..7)
Data Out a (0..17)
RD/WR a high
RD/WR a low
CS a
Reset
Busy a
Data In b (0..17)
Address b (0..7)
Data Out b (0..17)
RD/WR b high
RD/WR b low
CS b
Busy b
Full
Almost Full
Almost Empty
Empty
420
411b
411a
412a
410b
421
413a
414a
415a
417
403
404a
402
401
400
A 0
A 1
To PreMUX
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

FIG. 4a(i)

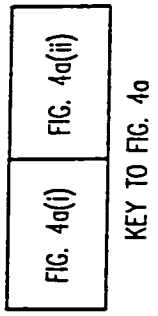


FIG. 4a(ii)
FIG. 4a(i)
KEY TO FIG. 4a

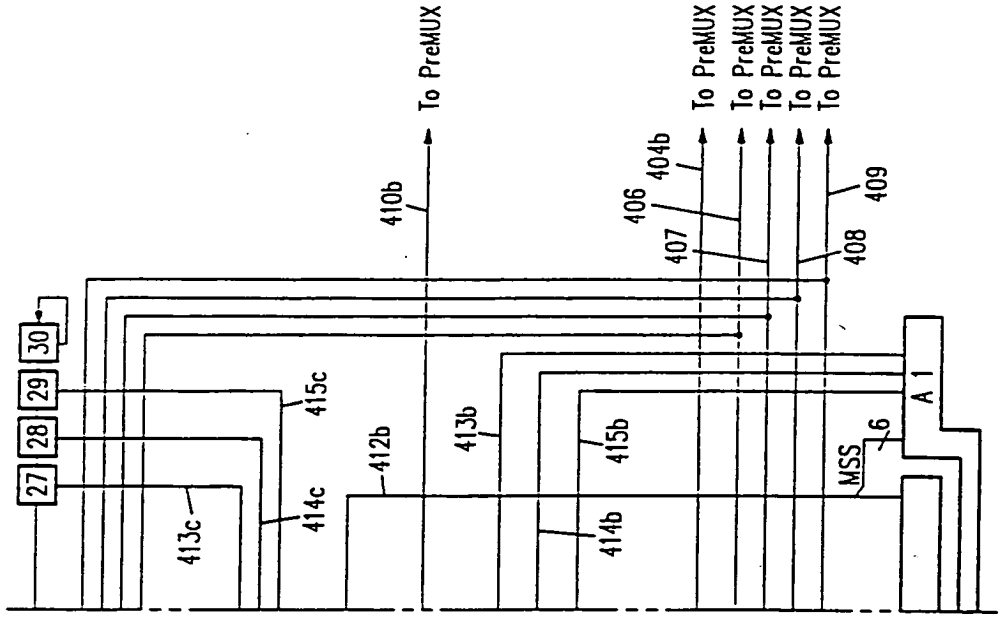

To PreMUX
To PreMUX
To PreMUX
To PreMUX
To PreMUX
To PreMUX
410b
404b
406
407
408
409
30
29
28
27
415c
414c
413c
412b
413b
415b
414b
A 1
6
MSS

FIG. 4a(ii)

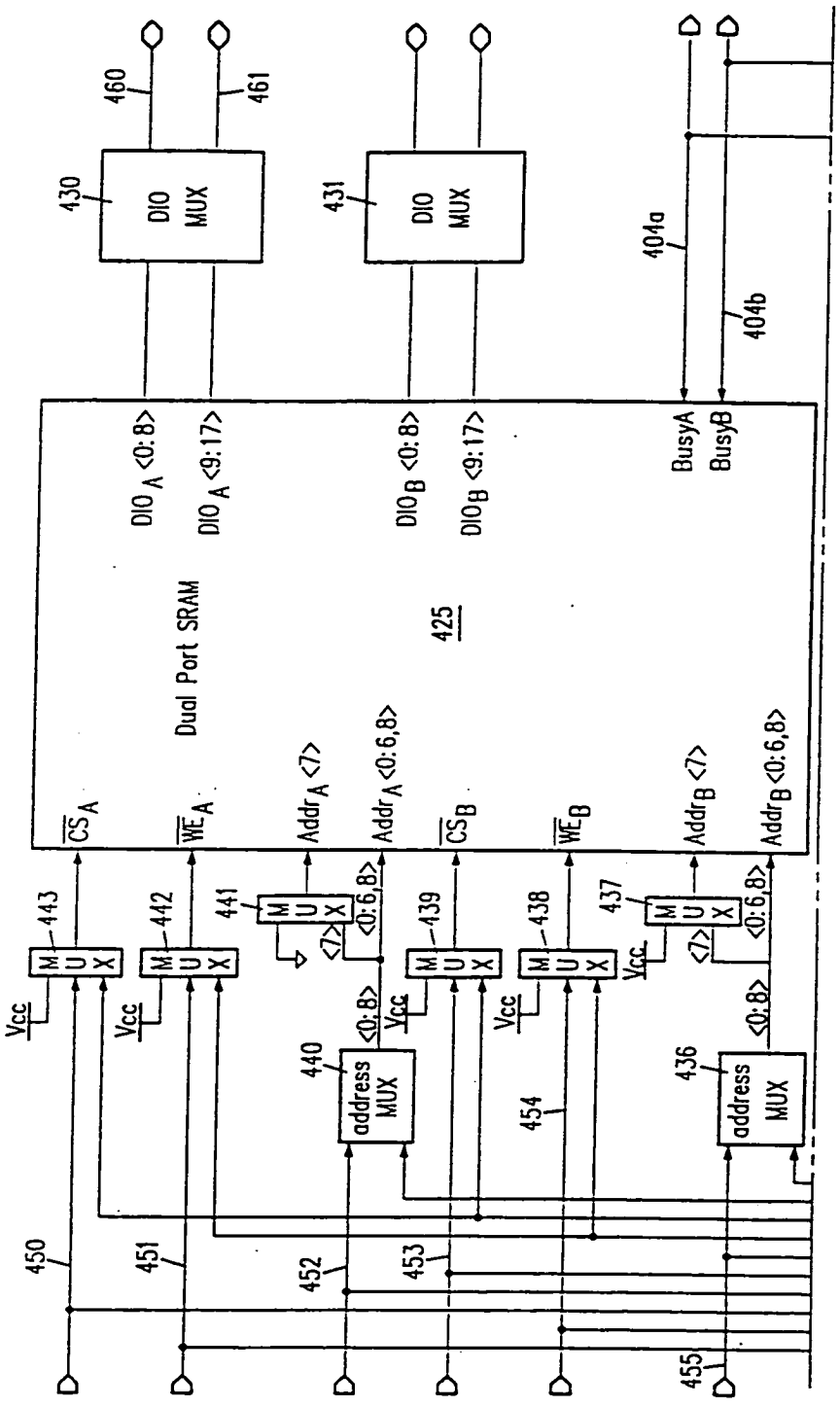

460
461
430
DIO MUX
431
DIO MUX
404a
404b
DIO A <0:8>
DIO A <9:17>
DIO B <0:8>
DIO B <9:17>
BusyA
BusyB
Dual Port SRAM
425
CS A
WE A
Addr A <7>
Addr A <0:6,8>
CS B
WE B
Addr B <7>
Addr B <0:6,8>
443
442
441
439
438
437
MUX
Vcc
<0:8>
<7>
<0:6,8>
440
436
address MUX
454
450
451
452
453
455

FIG. 4b(i)

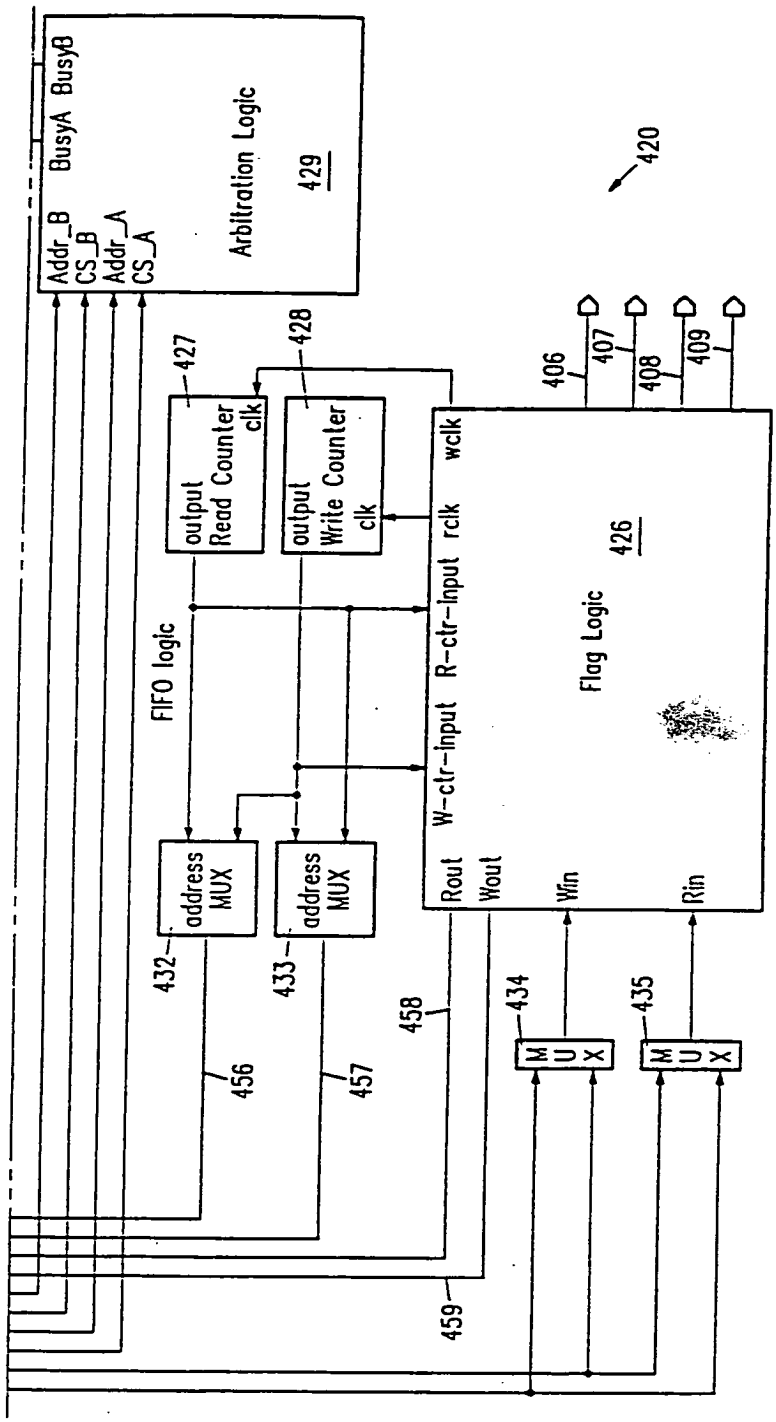

Arbitration Logic
429
BusyA
BusyB
Addr_B
CS_B
Addr_A
CS_A
420
FIG. 4b(i)
FIG. 4b(ii)
KEY TO FIG. 4b
427
428
406
407
408
409
output Read Counter
clk
output Write Counter
clk
FIFO logic
Flag Logic
426
wclk
rclk
W-ctr-input
R-ctr-input
Rout
Wout
Win
Rin
address MUX
address MUX
432
433
458
434
435
456
457
459
M U X
M U X

FIG. 4b(ii)

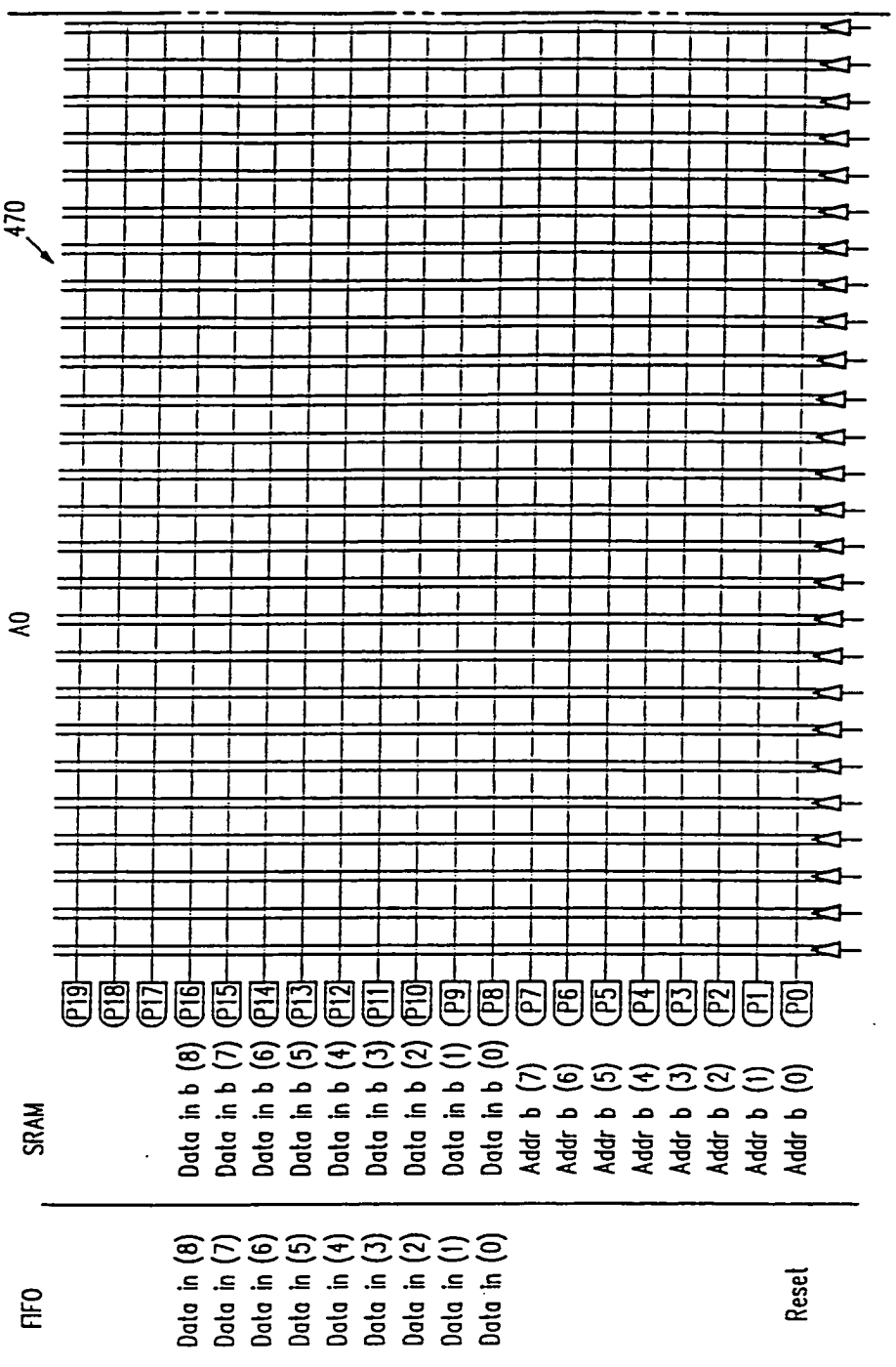

470
A0
SRAM
FIFO
P19
P18
P17
P16
P15
P14
P13
P12
P11
P10
P9
P8
P7
P6
P5
P4
P3
P2
P1
P0
Data in b (8)
Data in b (7)
Data in b (6)
Data in b (5)
Data in b (4)
Data in b (3)
Data in b (2)
Data in b (1)
Data in b (0)
Addr b (7)
Addr b (6)
Addr b (5)
Addr b (4)
Addr b (3)
Addr b (2)
Addr b (1)
Addr b (0)
Data in (8)
Data in (7)
Data in (6)
Data in (5)
Data in (4)
Data in (3)
Data in (2)
Data in (1)
Data in (0)
Resel
FIG. 4c(i)
FIG. 4c(ii)
KEY TO FIG. 4c

FIG. 4c(i)

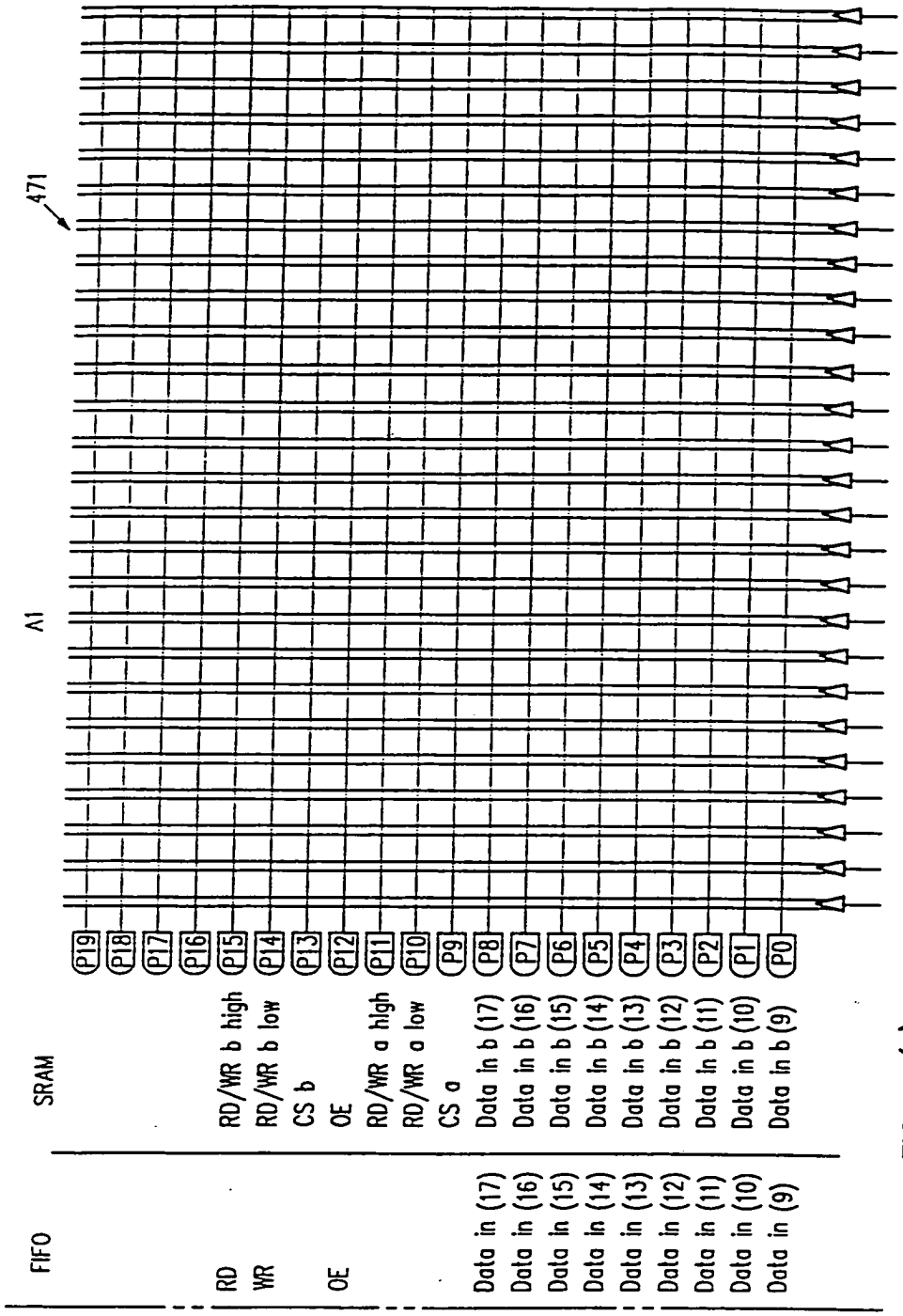

471
A1
P19
P18
P17
P16
P15
P14
P13
P12
P11
P10
P9
P8
P7
P6
P5
P4
P3
P2
P1
P0
SRAM
RD/WR b high
RD/WR b low
CS b
OE
RD/WR a high
RD/WR a low
CS a
Data in b (17)
Data in b (16)
Data in b (15)
Data in b (14)
Data in b (13)
Data in b (12)
Data in b (11)
Data in b (10)
Data in b (9)
FIFO
RD
WR
OE
Data in (17)
Data in (16)
Data in (15)
Data in (14)
Data in (13)
Data in (12)
Data in (11)
Data in (10)
Data in (9)

FIG. 4c(ii)

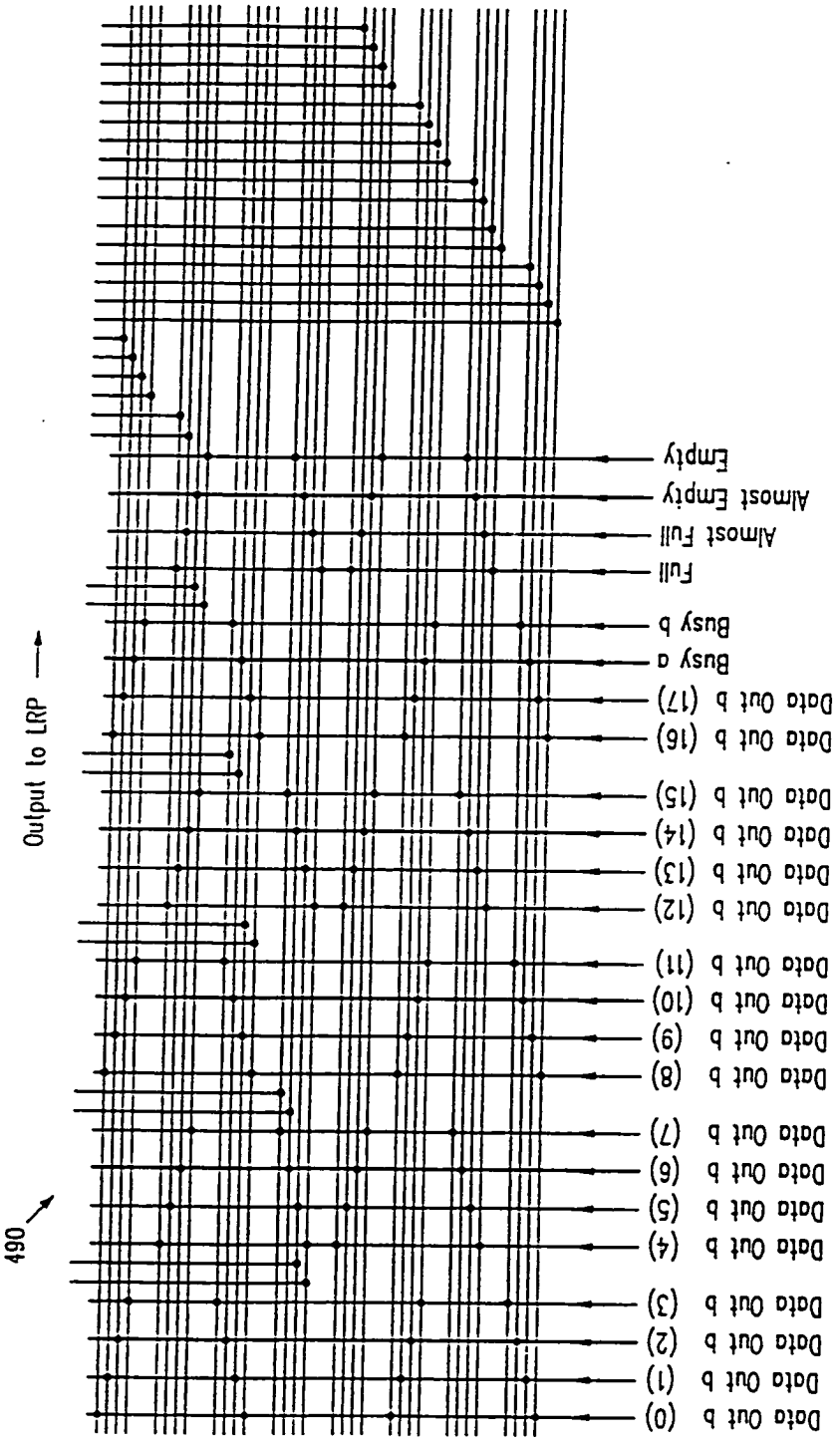

Output to LRP
490
Data Out b (0)
Data Out b (1)
Data Out b (2)
Data Out b (3)
Data Out b (4)
Data Out b (5)
Data Out b (6)
Data Out b (7)
Data Out b (8)
Data Out b (9)
Data Out b (10)
Data Out b (11)
Data Out b (12)
Data Out b (13)
Data Out b (14)
Data Out b (15)
Data Out b (16)
Data Out b (17)
Busy a
Busy b
Full
Almost Full
Almost Empty
Empty

FIG. 4d

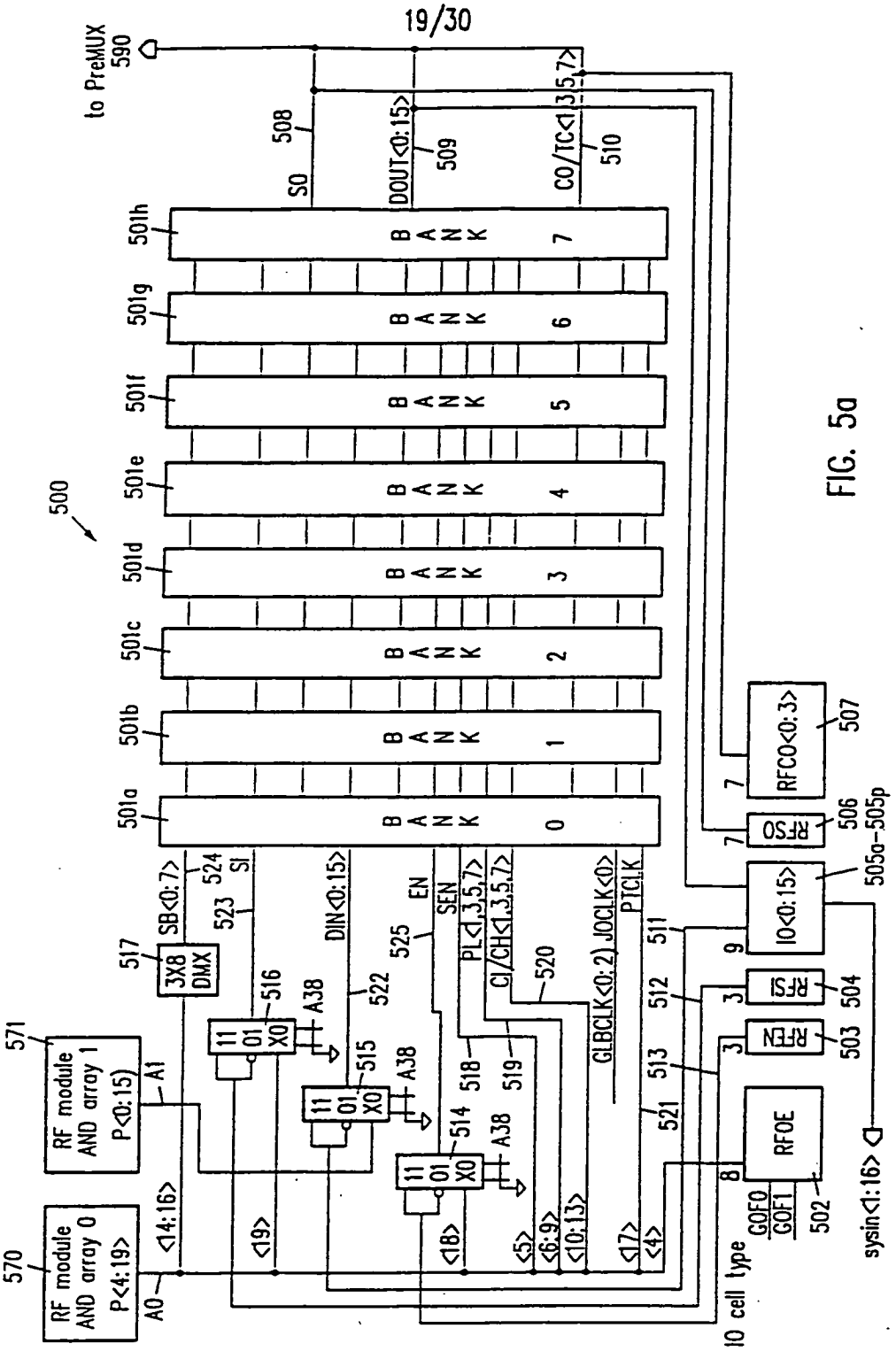

to PreMUX
590
500
501h
501g
501f
501e
501d
501c
501b
501a
BANK
7
6
5
4
3
2
1
0
SO
508
DOUT<0:15>
509
CO/TC<1,3,5,7>
510
RFCO<0:3>
507
RFSO
506
IO<0:15>
505a-505p
RFSI
504
RFEN
503
RFOE
502
GOE0
GOE1
IO cell type
sysin<1:16>
SB<0:7>
524
SI
523
3X8 DMX
517
DIN<0:15>
522
EN
525
SEN
PL<1,3,5,7>
CI/CH<1,3,5,7>
520
GLBCLK<0:2>
JOCLK<0>
PTCLK
511
512
513
521
516
515
514
518
519
A38
11
01
X0
571
RF module AND array 1 P<0:15>
A1
570
RF module AND array 0 P<4:19>
A0
<14:16>
<19>
<18>
<5>
<6:9>
<10:13>
<17>
<4>

FIG. 5a

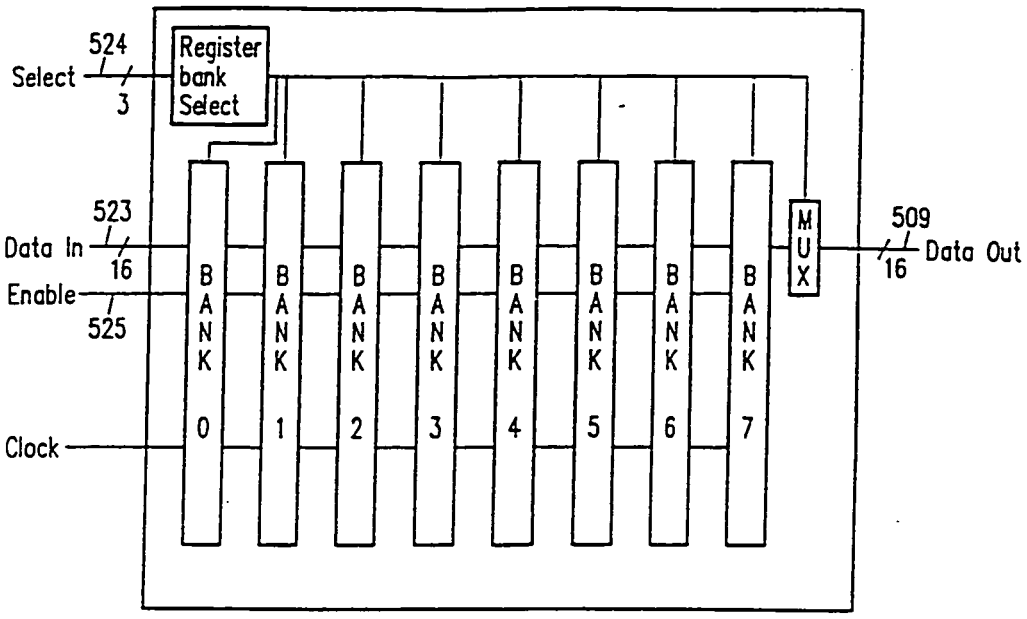


FIG. 5b(i)

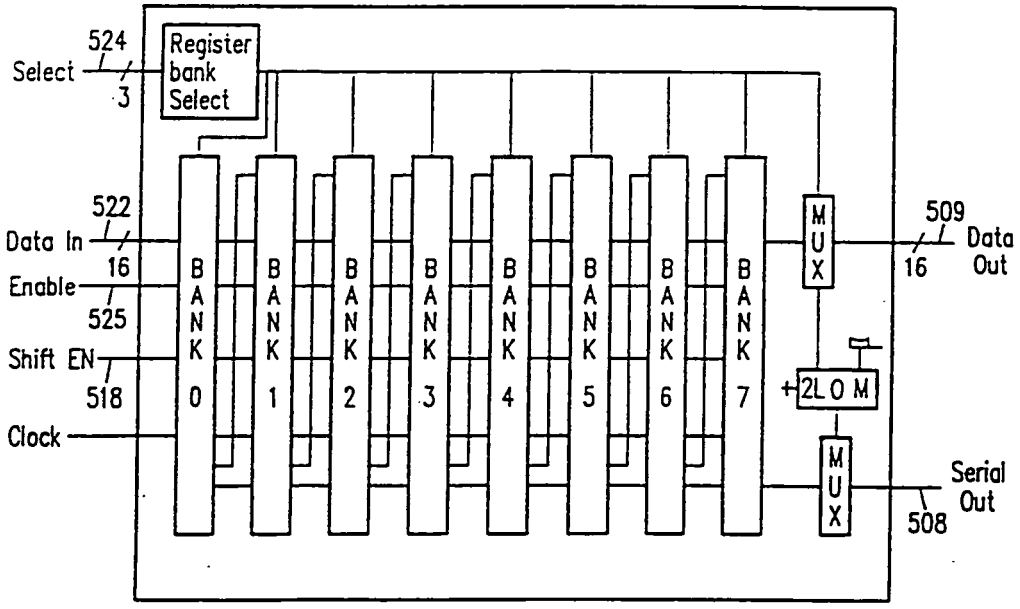


FIG. 5b(ii)

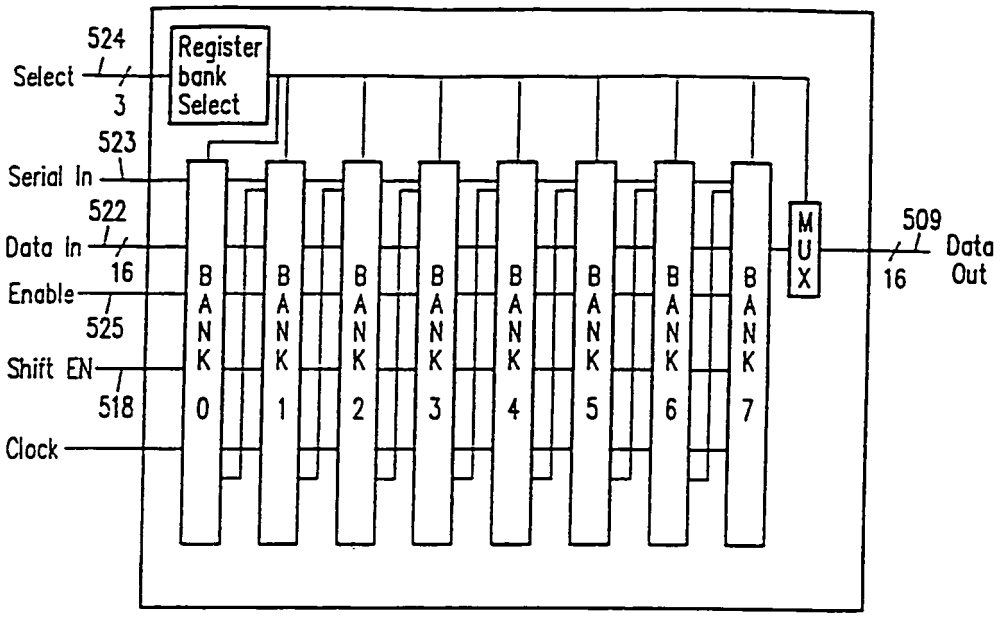


FIG. 5b(iii)

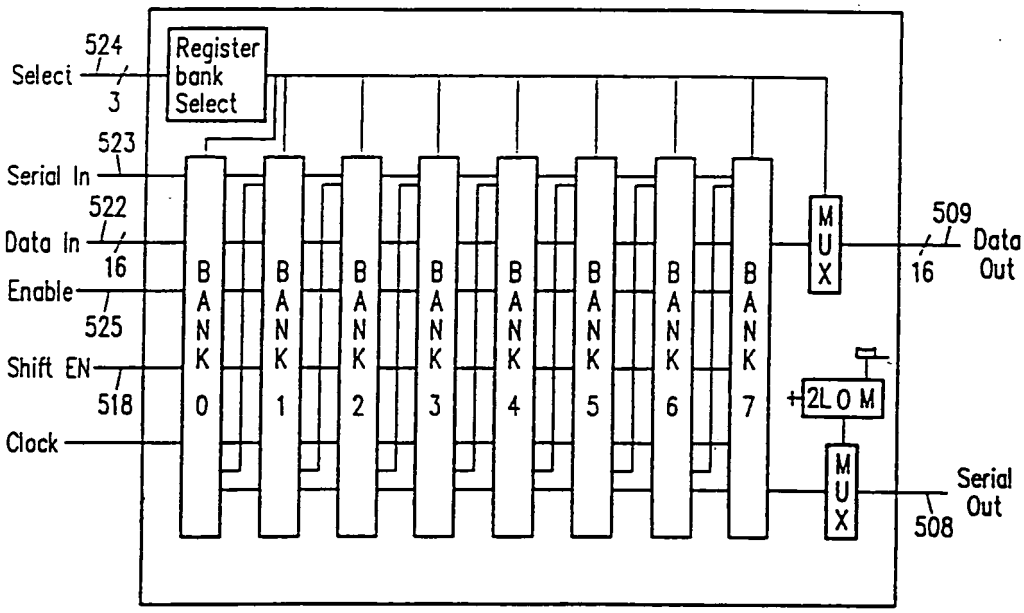


FIG. 5b(iv)

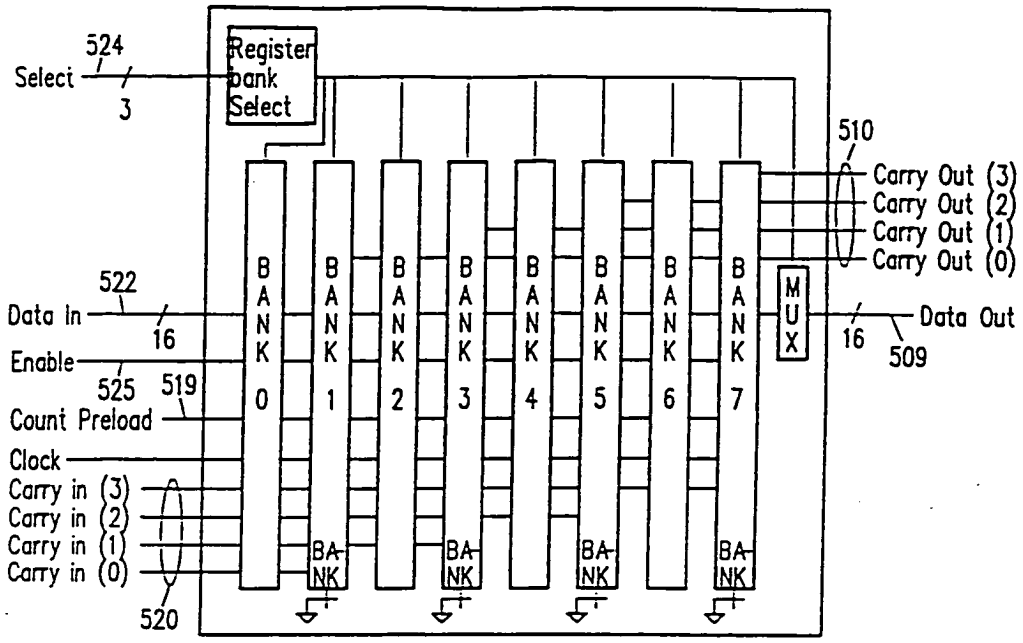


FIG. 5b (v)

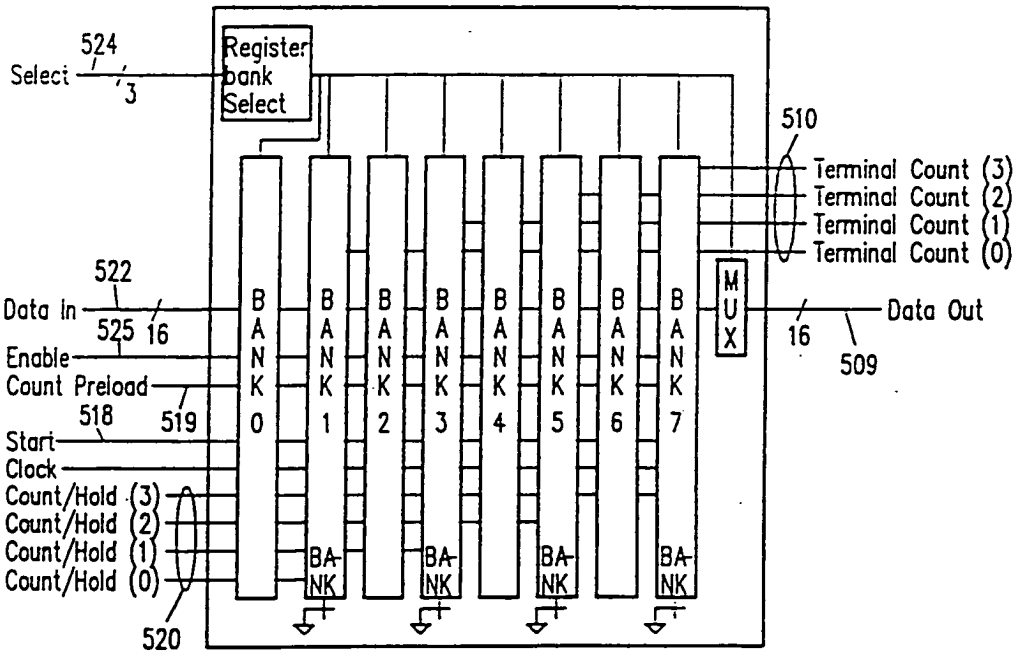
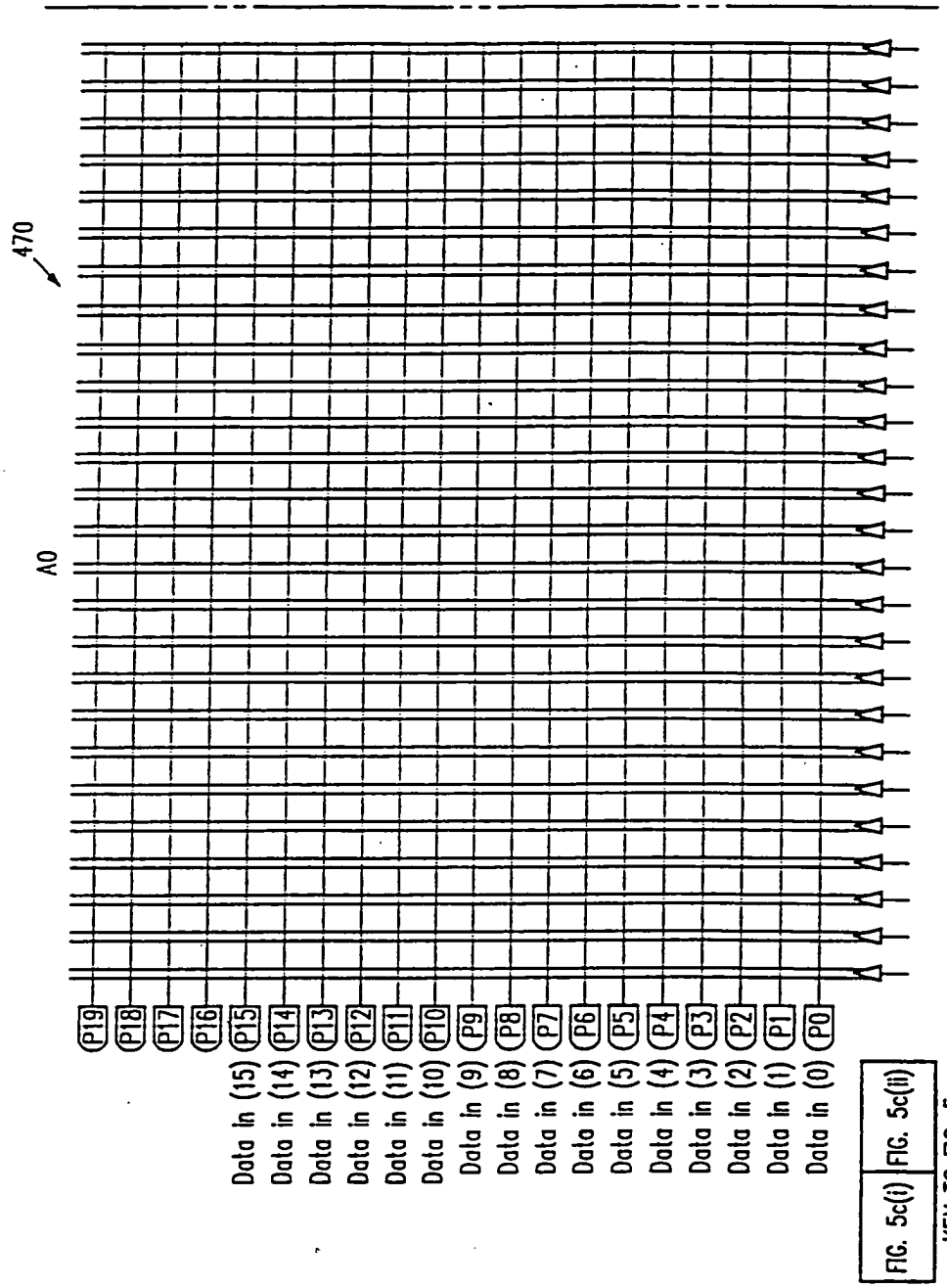


FIG. 5b (vi)

FIG. 5c(i)


470
A0
P19
P18
P17
P16
P15
P14
P13
P12
P11
P10
P9
P8
P7
P6
P5
P4
P3
P2
P1
P0
Data in (15)
Data in (14)
Data in (13)
Data in (12)
Data in (11)
Data in (10)
Data in (9)
Data in (8)
Data in (7)
Data in (6)
Data in (5)
Data in (4)
Data in (3)
Data in (2)
Data in (1)
Data in (0)
FIG. 5c(i)
FIG. 5c(ii)
KEY TO FIG. 5c

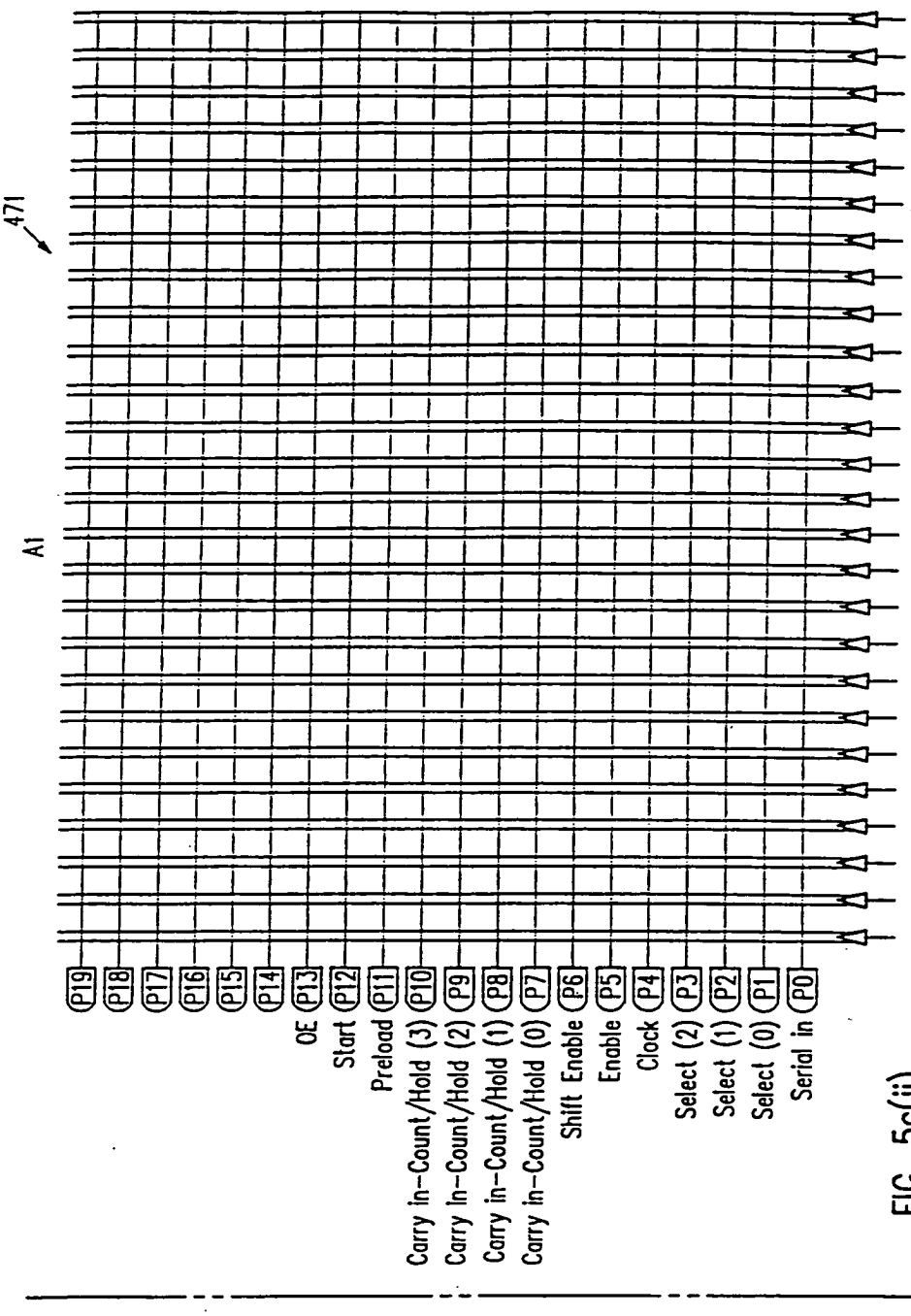

471
A1
P19
P18
P17
P16
P15
P14
P13
P12
P11
P10
P9
P8
P7
P6
P5
P4
P3
P2
P1
P0
OE
Start
Preload
Carry in-Count/Hold (3)
Carry in-Count/Hold (2)
Carry in-Count/Hold (1)
Carry in-Count/Hold (0)
Shift Enable
Enable
Clock
Select (2)
Select (1)
Select (0)
Serial in

FIG. 5c(ii)

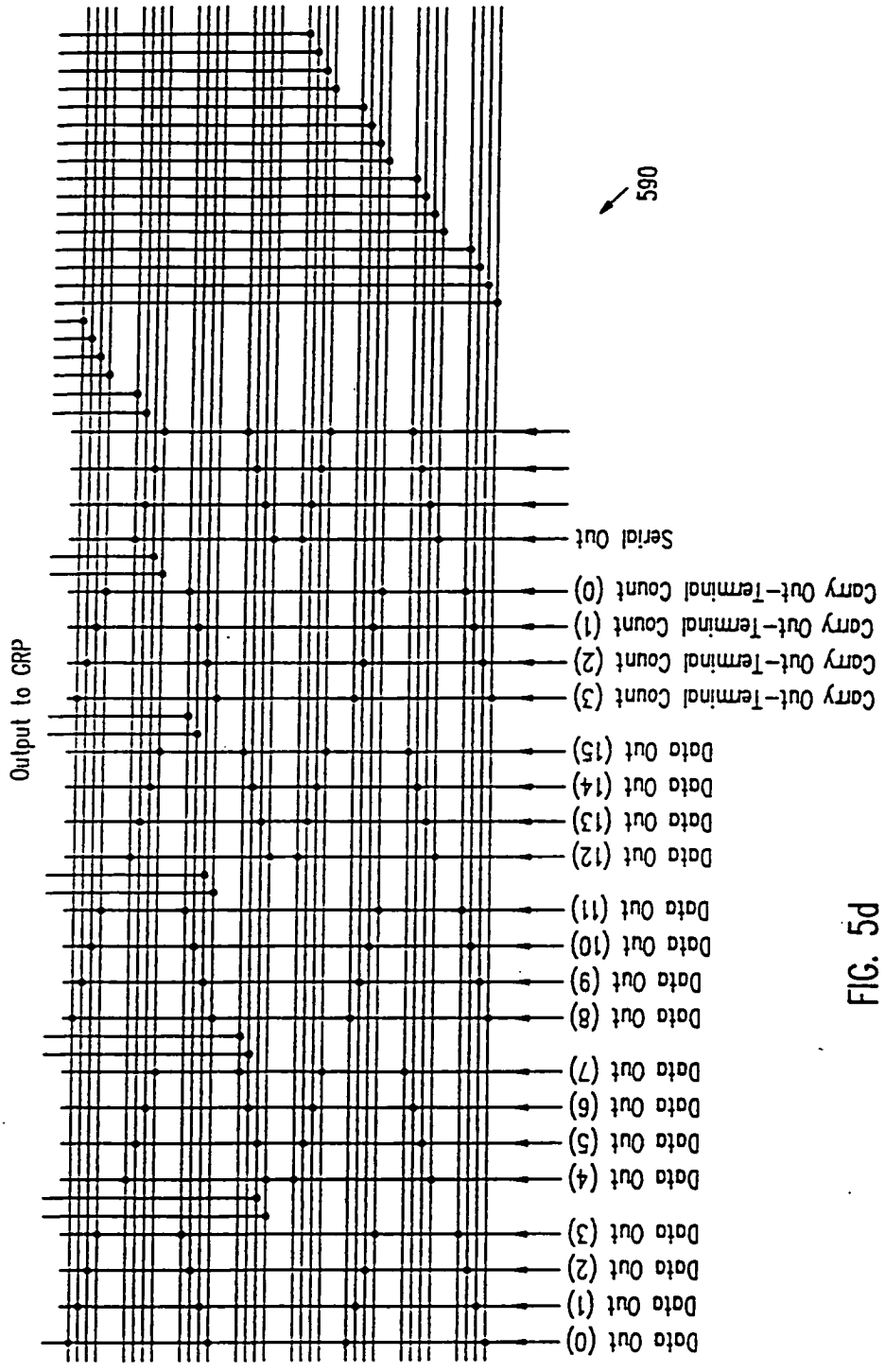

590
Output to GRP
Serial Out
Carry Out–Terminal Count (0)
Carry Out–Terminal Count (1)
Carry Out–Terminal Count (2)
Carry Out–Terminal Count (3)
Data Out (15)
Data Out (14)
Data Out (13)
Data Out (12)
Data Out (11)
Data Out (10)
Data Out (9)
Data Out (8)
Data Out (7)
Data Out (6)
Data Out (5)
Data Out (4)
Data Out (3)
Data Out (2)
Data Out (1)
Data Out (0)

FIG. 5d

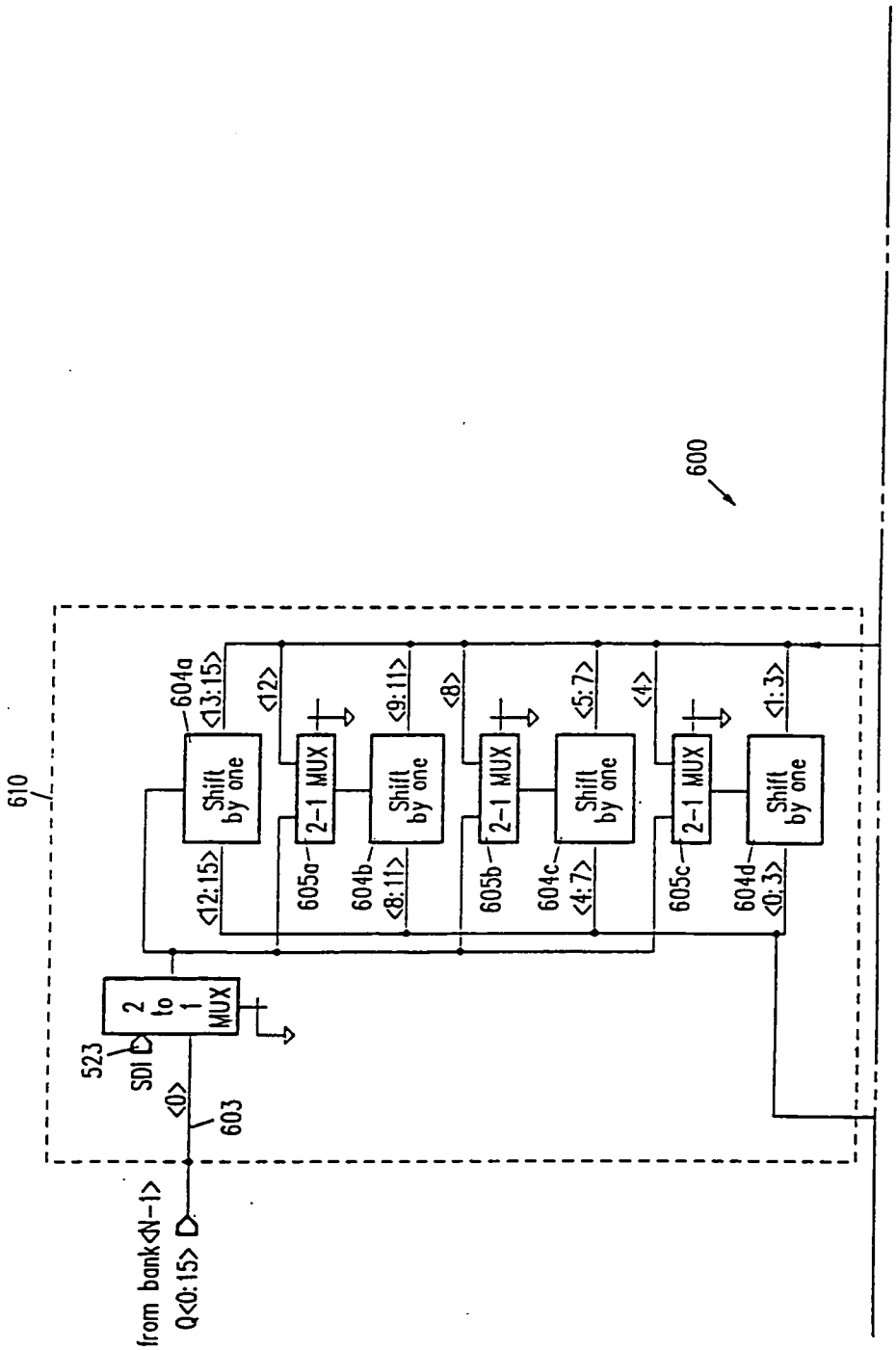

600
610
604a
<13:15>
<12>
<9:11>
<8>
<5:7>
<4>
<1:3>
Shift by one
2-1 MUX
Shift by one
2-1 MUX
Shift by one
2-1 MUX
Shift by one
<12:15>
605a
604b
<8:11>
605b
604c
<4:7>
605c
604d
<0:3>
2 to 1 MUX
523
SDI
<0>
603
from bank<N-1>
Q<0:15>

FIG. 6a(i)

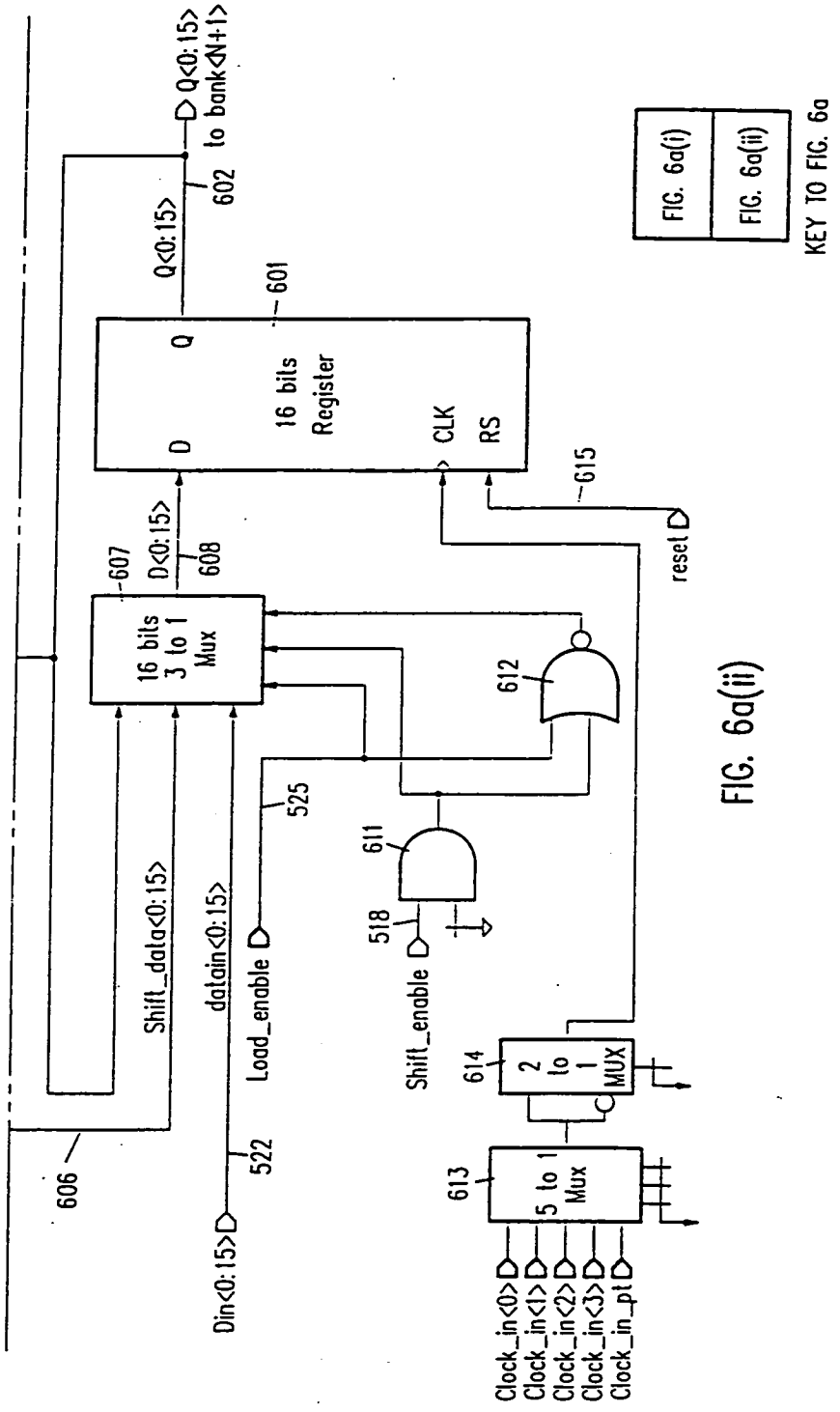

Q<0:15> to bank<N+1>
602
Q<0:15>
601
16 bits Register
D
Q
CLK
RS
615
reset
D<0:15>
608
607
16 bits 3 to 1 Mux
612
525
611
518
Shift_enable
Load_enable
datain<0:15>
Shift_data<0:15>
606
522
Din<0:15>
614
2 to 1 MUX
613
5 to 1 Mux
Clock_in<0>
Clock_in<1>
Clock_in<2>
Clock_in<3>
Clock_in_pt
FIG. 6a(i)
FIG. 6a(ii)
KEY TO FIG. 6a

FIG. 6a(ii)

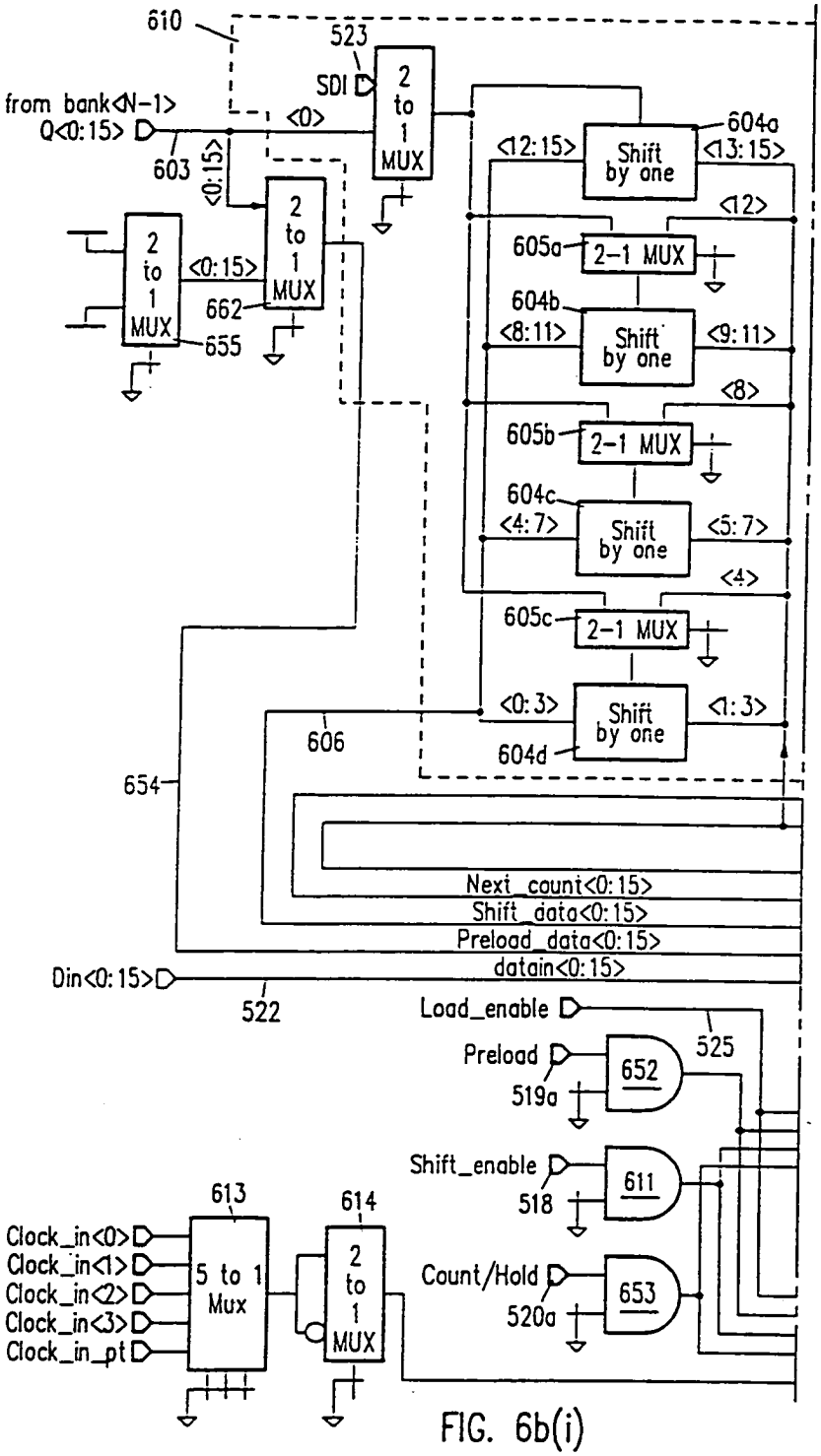

610
523
SDI
2 to 1 MUX
from bank<N-1> Q<0:15>
<0>
603
<0:15>
2 to 1 MUX
<0:15>
662
655
<12:15>
Shift by one
604a
<13:15>
<12>
605a
2-1 MUX
604b
<8:11>
Shift by one
<9:11>
<8>
605b
2-1 MUX
604c
<4:7>
Shift by one
<5:7>
<4>
605c
2-1 MUX
<0:3>
Shift by one
<1:3>
606
604d
654
Next_count<0:15>
Shift_data<0:15>
Preload_data<0:15>
datain<0:15>
Din<0:15>
522
Load_enable
525
Preload
519a
652
Shift_enable
518
611
Count/Hold
520a
653
613
614
Clock_in<0>
Clock_in<1>
Clock_in<2>
Clock_in<3>
Clock_in_pt
5 to 1 Mux
2 to 1 MUX

FIG. 6b(i)

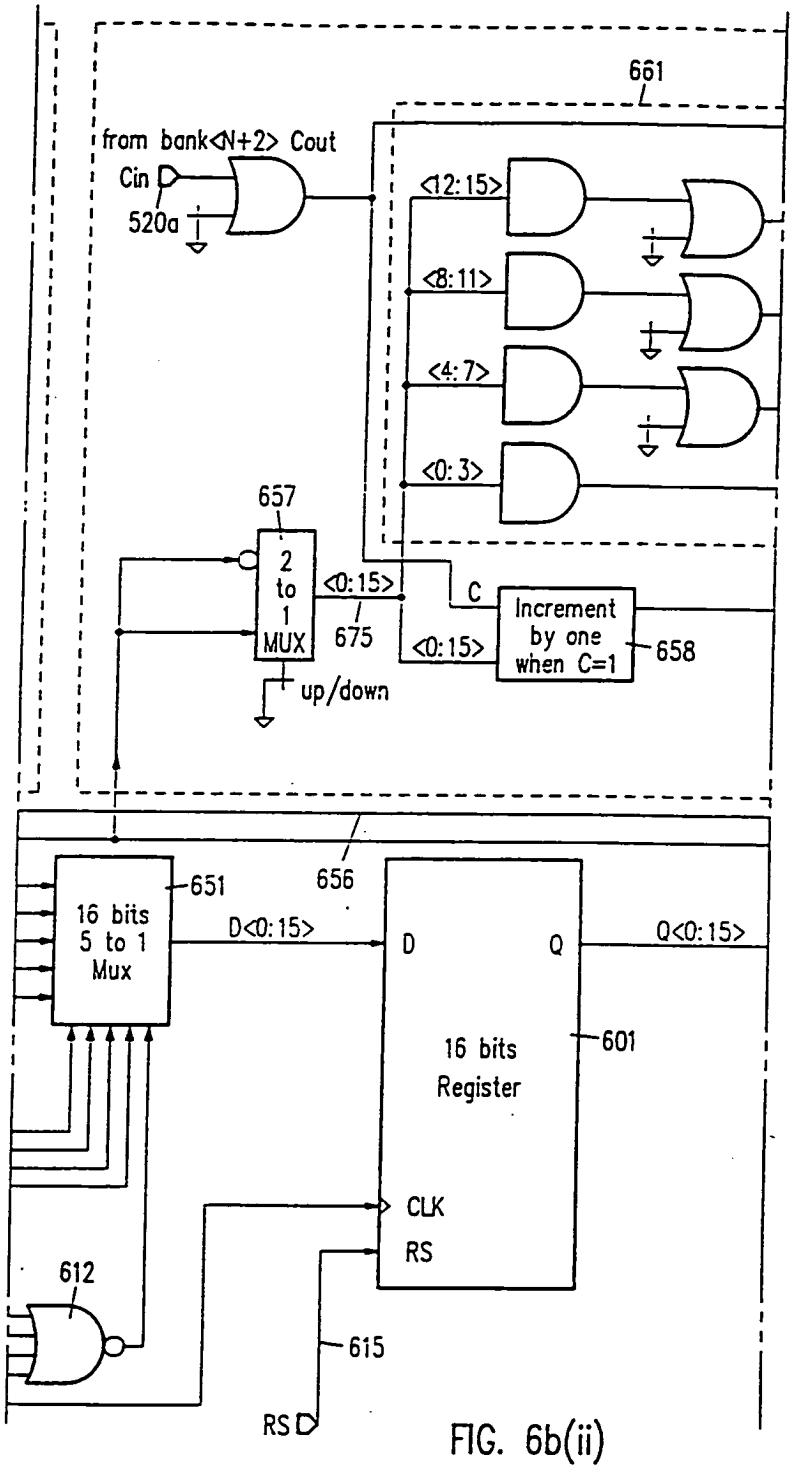

661
from bank<N+2> Cout
Cin
520a
<12:15>
<8:11>
<4:7>
<0:3>
657
2 to 1 MUX
<0:15>
675
C
Increment by one when C=1
658
<0:15>
up/down
656
651
16 bits 5 to 1 Mux
D<0:15>
D
Q
Q<0:15>
16 bits Register
601
CLK
RS
612
615
RS

FIG. 6b(ii)

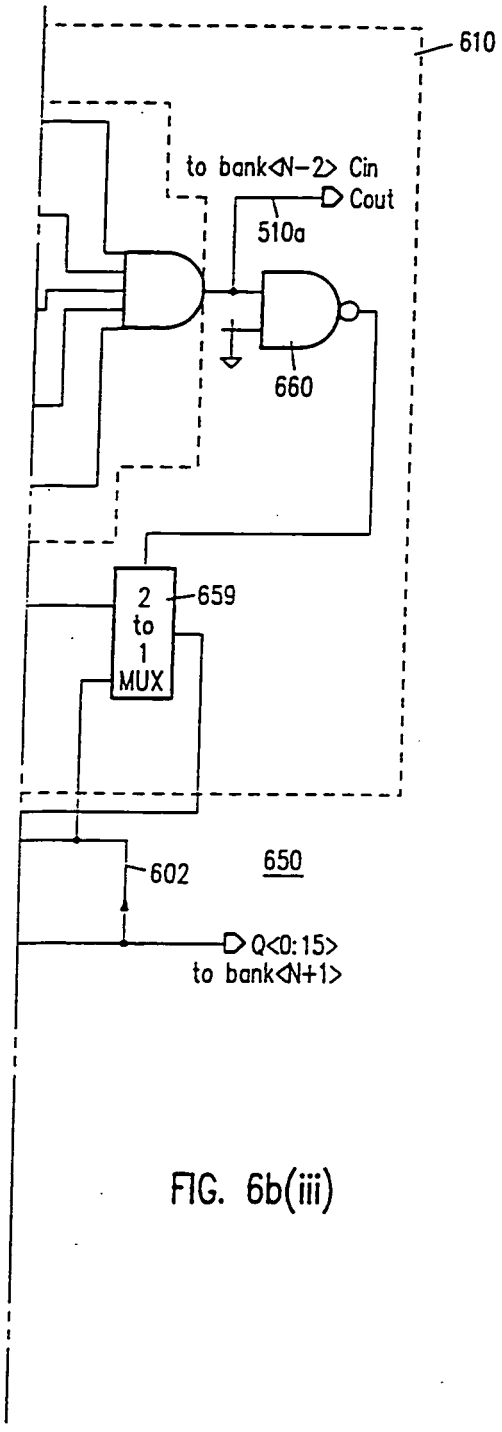


FIG. 6b(iii)

FIG. 6b(i)	FIG. 6b(ii)	FIG. 6b(iii)

KEY TO FIG. 6b